Alternative Discrete-Time Operators and Their Application to Nonlinear Models

Andrew D. Back[†]
Laboratory for Artificial Brain Systems,
Frontier Research Program RIKEN,
The Institute of Physical and Chemical Research,
2-1 Hirosawa, Wako–shi,
Saitama 351-01, Japan

Ah Chung Tsoi[†]
Faculty of Informatics
University of Wollongong
Northfields Avenue, Wollongong
Australia

Bill G. Horne
AADM Consulting
9 Pace Farm Rd.
Califon, NJ 07830 USA

C. Lee Giles[*]
NEC Research Institute
4 Independence Way
Princeton, NJ 08540. USA

Technical Report CS-TR-3738 and UMIACS-TR-97-03
Institute for Advanced Computer Studies
University of Maryland, College Park, Md 20742

[†]The authors were previously with the Department of Electrical and Computer Engineering, University of Queensland, Brisbane Qld. 4072 Australia.
*Also with Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 USA.

Abstract

The shift operator, defined as q x(t) = x(t+1), is the basis for almost all discrete-time models. It has been shown however, that linear models based on the shift operator suffer problems when used to model lightly-damped-low-frequency (LDLF) systems, with poles near $(1,0)$ on the unit circle in the complex plane. This problem occurs under fast sampling conditions. As the sampling rate increases, coefficient sensitivity and round-off noise become a problem as the difference between successive sampled inputs becomes smaller and smaller. The resulting coefficients of the model approach the coefficients obtained in a binomial expansion, regardless of the underlying continuous-time system. This implies that for a given finite wordlength, severe inaccuracies may result. Wordlengths for the coefficients may also need to be made longer to accommodate models which have low frequency characteristics, corresponding to poles in the neighbourhood of (1,0). These problems also arise in neural network models which comprise of linear parts and nonlinear neural activation functions. Various alternative discrete-time operators can be introduced which offer numerical computational advantages over the conventional shift operator. The alternative discrete-time operators have been proposed independently of each other in the fields of digital filtering, adaptive control and neural networks. These include the delta, rho, gamma and bilinear operators. In this paper we first review these operators and examine some of their properties. An analysis of the TDNN and FIR MLP network structures is given which shows their susceptibility to parameter sensitivity problems. Subsequently, it is shown that models may be formulated using alternative discrete-time operators which have low sensitivity properties. Consideration is given to the problem of finding parameters for stable alternative discrete-time operators. A learning algorithm which adapts the alternative discrete-time operators parameters on-line is presented for MLP neural network models based on alternative discrete-time operators. It is shown that neural network models which use these alternative discrete-time perform better than those using the shift operator alone.

Keywords: Shift operator, alternative discrete-time operator, gamma operator, rho operator, low sensitivity, time delay neural network, high speed sampling, finite wordlength, LDLF, MLP, TDNN.

1 Introduction

Recent interest has concentrated in deriving various neural network architectures, often based on a modification of the classic multilayer perceptron (MLP) [22] for nonlinear functional mapping approximations, for modelling time-dependent signals. For example, a popular architecture, commonly known as the time delay neural network (TDNN) model [27, 58], is based on the MLP, except that the input signal to each node (input or hidden) can include delayed versions of the same signal. It is known that this method works quite well in a number of applications, e.g., speech processing [58]. However, it is a common observation that this model may be computationally expensive.
One way in which the TDNN model can be understood is as follows:
An MLP model having L layers with $N_0, N_1, ..., N_L$ nodes per layer, is defined

$$z_k^l(t) = f\left(\hat{x}_k^l(t)\right) \tag{1}$$

$$\hat{x}_k^l(t) = \begin{cases} \sum_{i=1}^{N_l} w_{ki}^l z_i^{l-1}(t) & l = 2, ..., L \\ w_k^1 x(t) & l = 1 \end{cases} \tag{2}$$

where $x(t)$ is the input to the system (for clarity only a single input is shown), each neuron i in layer l has an output of $z_i^l(t)$; a layer consists of N_l neurons ($l = 0$ denotes the input layer, and $l = L$ denotes the output layer, $z_{N_l}^l = 1.0$ may be used for a bias); $f(\cdot)$ is a sigmoid function typically evaluated as $\tanh(\cdot)$, and a synaptic connection between unit i in the previous layer and unit k in the current layer is represented by w_{ki}^l. The notation t may be used to represent a discrete time or pattern instance.
The TDNN model is obtained by augmenting the input $x(t)$ and $z_i^{l-1}(t)$ as follows:

$$\upsilon = \left[\begin{array}{cccc} x(t) & x(t-1) & \dots & x(t-\kappa+1) \end{array} \right]' \tag{3}$$

and

$$\zeta = \left[\begin{array}{cccc} z_i^{l-1}(t) & z_i^{l-1}(t-1) & \dots & z_i^{l-1}(t-\ell_i+1) \end{array} \right]' \tag{4}$$

where $l = 2, \dots, L$. $\sum_{i=1}^{n} \ell_i = \lambda$. υ and ζ are respectively κ and λ dimensional vectors. There are correspondingly more weights w_{ki}^l associated with each signal $x(t-i)$, or z_i^{l-1}. The notation $'$ denotes the transpose of a vector.
Note that if we introduce a shift operator $q^{-1}z(t) \triangleq z(t-1)$, then equation (3) can be written equivalently as follows

$$\upsilon = \left[\begin{array}{cccc} 1 & q^{-1} & \dots & q^{-\kappa+1} \end{array} \right]' \mathbf{x(t)} \tag{5}$$

Equation (5) is sometimes referred to as the regression vector in the literature [21]. Equation (4) can also be expressed in a regression vector form quite easily.
It is noted that the performance of the TDNN as a signal model for time varying signals is dependent on the parameters κ and λ, as well as the weights. The parameters κ and λ express the dependence of the output signal on its past data. Hence, if κ or λ is large, it indicates that there may be long term dependence on the past signals.
Equation (3) can also be interpreted in terms of tapped delay lines. Thus, the input signal $x(t)$ is passed through a tapped delay line of order κ. Similarly, the output of the ith hidden layer neuron $z_i^{l_i-1}$ can also be interpreted as passing through delay lines of length ℓ_i.
It is possible to have other neural network architectures, based on a modification of the TDNN model. For example, instead of having just a tapped delay line at the input of each neuron, it is possible to incorporate a finite impulse response (FIR) filter[1]. It is well known that a long FIR

[1] An FIR filter is the same as a tapped delay line, except that each of the taps in the delay line is multiplied by a weight. It is a re-interpretation of the TDNN model, i.e., without any additional parameters.

filter can be modelled more efficiently (in terms of the number of parameters used) by using an infinite impulse response (IIR) filter. For convenience, we will refer to these models as FIR MLP and IIR MLP respectively [3, 59]. Other dynamic neural network structures have been proposed in the literature which involve some form of time delayed inputs, outputs, or hidden states (see for example [8, 12, 15, 18, 20, 25, 36, 37, 43]). In this paper we will consider feedforward neural network structures which have an FIR sub-structure, but the results are generalizable to these more complex network architectures. The implications of this work for recurrent neural networks will be considered in another paper.

The shift operator q works well in practice. However, there are situations in which it does not work too well. One of the conditions under which it does not work well is commonly known as the lightly damped low frequency (LDLF) condition. This condition can be simplest understood from the sensitivity point of view.

Consider the following polynomial which may be associated with an FIR or IIR synapse:

$$A(q) = 1 + a_1 q^{-1} + \cdots + a_n q^{-n} \tag{6}$$

$$= q^{-n} \prod_{i=1}^{n} (q - \lambda_i) \tag{7}$$

where λ_i is the ith root of the polynomial equation (6). The sensitivity of this polynomial is defined as [32]

$$S_{ij} = \frac{d\lambda_i}{da_j}$$

$$= \frac{\lambda_i^{n-j}}{\prod_{k \neq i}(\lambda_i - \lambda_k)} \tag{8}$$

It is observed in equation (8), there are two conditions under which the sensitivity is high:

1. $\lambda_i \approx \lambda_k$, $i \neq k$. This indicates that the roots are close to one another.
2. λ_i has a large magnitude. Now, for the polynomial equation (6) to be stable, the largest possible magnitude for the roots will be ≈ 1.

There are a number of practical situations under which these conditions can occur. These include:

1. Fast sampling – In sampling a continuous time system, if the sampling frequency is progressively higher, well beyond the natural frequencies of the underlying continuous time system, then there is progressively less and less information contained in neighbouring samples.
2. LDLF models – This situation is similar to fast sampling, where the poles and/or zeros of the system are concentrated around $(1, 0)$ on the complex plane, i.e., the roots are close to one another and they all have magnitude ≈ 1.

In the digital implementation of a model, because of the hardware limitations, it is possible that, while a parameter is represented ideally in infinite precision (or at least of sufficient precision that the finite precision effects are not noticeable, with respect to the signal magnitude), a finite wordlength implementation may introduce inaccuracies. Here, a model with high sensitivity will have a large change in behaviour due to small changes in the parameters resulting from finite precision implementations. Hence, the condition of fast sampling or modelling LDLF systems cannot be well handled by shift operator models which use a finite precision implementation.

There are two common methods to overcome these potential difficulties.

1. Subsampling – since the neighbouring samples do not contain sufficient information (i.e. they contain much redundant information), one way in which neighbouring samples can be made to contain more information is by taking only one sample out of, say, N samples.

2. Low pass filtering – if a signal is sampled often, it may result in a high frequency noise being imposed on the sampled signal. Thus one way in which this can be overcome is by the incorporation of a low pass filter, with the explicit aim to filter out the underlying low frequency signals.

In this paper, we will consider only the "low pass" filtering operations, and we will not consider the sub sampling method further.
In general, the design of the low pass filter presumes that a cut off frequency is known. A simple method in which the low pass filtering can be accomplished is by the employment of what is commonly known as alternative discrete time operators (ADTOs).
The principle of operation of the ADTOs is quite simple. What one desires to do is to find an alternative operator ν, such that the sensitivity of the original model is reduced.
Consider the original model $A(q)$ (see equation (6)). We seek an alternative operator ν such that the transformed model

$$A_t(\nu) = 1 + a_1\nu^{-1} + \ldots + a_n\nu^{-n} \tag{9}$$

has a reduced sensitivity. The alternative operator ν is commonly obtained as a nonlinear transformation of the shift operator q, i.e., $\nu = g(q)$, where $g(\cdot)$ is a nonlinear transformation.
In the design of the nonlinear function $g(\cdot)$, a guiding principle is that it must be a low pass filter.
There are a number of possible ADTOs which have been proposed in the signal processing and neural network literature. To our knowledge, there has not been a unified treatment of these ADTOs. Thus, in this paper, we will first give a unified treatment of these ADTOs, indicating how they relate to one another. In addition, we will consider their stability regions which is important in their design.
Secondly, we turn our attention to the following questions:

1. Does the phenomenon of "high sensitivity" occur in neural network architectures, based on a modification of the MLP structure, e.g., TDNN, FIR MLP?
2. If this phenomenon occurs, can its effect be reduced by the deployment of ADTOs?

Note that some care needs to be taken with the consideration of the phenomenon of high sensitivity problems in neural networks. Clearly, the different structure of an FIR MLP for example, means that we expect the resulting behaviour to be different from a linear polynomial (as an FIR MLP incorporates neurons in its structure). However, since the neural network structures we consider, such as the FIR MLP, contain linear filters as subsystems within the neural network structure, we will consider the problem in a similar manner as the linear models, even though we recognize that the manifestation of problems due to high sensitivity in each case may be significantly different.
Note further that, in this paper, we are not considering the important question of sampling a continuous time nonlinear system, e.g., the sampling of a continuous time Hopfield network. This is an open question, i.e., what are the effects of fast sampling on a continuous time neural network, with respect to the high sensitivity situation ?
In this paper, we do not claim to have found a complete formulation of the problem, nor do we claim that the solution proposed in the case of neural networks is necessarily optimal in terms of model sensitivity. What we claim is to have demonstrated, that the high sensitivity situation occurs in a special class of neural networks, that is, those which employ linear dynamic subsystems. Such models include the TDNN and those based on a modification of the synapses[2] of the classic MLP architecture, e.g., the FIR MLP, and that its effects can be more detrimental than in the case for linear systems. In addition, we also claim that in the limited manner which we have considered the problem, ADTOs appear to be capable of overcoming the problem of high sensitivity in the class of neural networks considered (eg. TDNN, FIR MLP). Thus, our results highlight the problem of high

[2]This is in contrast with modification of the underlying structure of the MLP, e.g., Elman model [15]. For further discussion on the differences between these two types of models, see, for example, the survey paper [57].

sensitivity in neural network modelling and indicate a solution which seems to be a promising area of further investigation.
The organisation of this paper is as follows: in section 2, we will consider briefly, the various common ADTOs which have been introduced. In section 3, we will consider the stability regions of these ADTOs. In section 4, we will consider briefly linear models and the application of ADTOs to these models. This is intended to provide as background on the treatment of the neural network models to follow. In section 5, we will introduce a special class of neural network models, designed to investigate the questions of ADTOs. An analysis of the neural network structures based on the shift operator is given where it shown that they are susceptible to parameter sensitivity problems. Subsequently, it is shown that models may be formulated using alternative discrete-time operators which have low sensitivity properties. We will give the derivation of a training algorithm as well as a number of observations concerning the application of ADTOs to this special class of neural networks in section 7. Finally, in section 8, we will present validation of some of the observations made in section 5 through a number of experiments. From the experiments, some observations and conclusions are drawn in section 9.

2 Alternative Discrete Time Operators

There have been a number of ADTOs proposed by various researchers. These include the following:

1. Delta operator.

   This is defined as [1]

   $$\delta = \frac{q-1}{\Delta} \tag{10}$$

   where Δ is the discrete-time sampling interval.

   Agarwal and Burrus first proposed the use of this operator in digital filters to replace the shift operator in an attempt to overcome the LDLF problems [1]. Williamson showed that the delta operator allows better performance in terms of coefficient sensitivity for digital filters derived from the direct form structure [61], and a number of authors have considered using it in linear filtering, estimation and control [16, 33, 42].

2. The modified delta operator.

   The delta operator can be modified to ensure that it is not marginally stable. The modified delta operator can be defined as

   $$\delta_m = \frac{q-c}{\Delta} \tag{11}$$

   where $0 < c < 1$.

3. Gamma operator.

   This operator is defined as [11]

   $$\gamma = \frac{q-(1-c)}{c} \tag{12}$$

   This is a generalization of the delta operator with the adjustable parameter c. The explicit sampling time Δ variable is implicitly set to one or incorporated in the variable c.

   This operator was originally introduced in [11, 13] as a means of studying neural network models for processing time-varying patterns.

4. Second order gamma operators.

   It is possible to introduce complex poles and zeros into the γ operator [10]. This operator is defined as

$$\gamma_2 = \frac{c_1 [z - (1 - c_1)]}{[z - (1 - c_1)]^2 + c_2 c_1^2} \tag{13}$$

   where c_i, $i = 1, 2$ are parameters of the γ_2 operator. The results in [10] indicate some success with this method, however, it was observed that a multimodal mean square error surface may occur in some modeling situations.

   In the γ_2 operator proposed in [10], while there may be a complex pole pair within the unit circle (if $(1 - c_1)^2 + c_1^2 c_2 < 1$), there is only one zero which lies on the real axis. This means that the operator is only capable of producing either low pass (for $0 < c_1 < 1$), or high pass filtering characteristics ($1 < c_1 < 2$). More general second order operators and their properties have been considered in [6].

5. Rho operator.

   The rho operator is defined as

$$\rho = \frac{q - (1 - c_1 \Delta)}{c_2 \Delta} \tag{14}$$

   where c_1, c_2 are adjustable parameters. The rho operator generalizes the delta and gamma operators. For the case where $c_1 \Delta = c_2 \Delta = 1$, the rho operator reduces to the usual shift operator . When $c_1 = 0$, and $c_2 = 1$, the rho operator reduces to the delta operator [41]. For $c_1 \Delta = c_2 \Delta = c$, the rho operator is equivalent to the gamma operator .

   This operator was introduced in the context of robust adaptive control [40]. It has been shown useful improvements over the performance of the delta operator [40, 41] can be obtained.

   One advantage of the rho operator over the delta operator is that it is stably invertible, allowing the derivation of simpler algorithms [40]. The ρ operator can be considered as a stable low pass filter, and parameter estimation using the ρ operator is low frequency biased. For adaptive control systems, this gives robustness advantages for systems with unmodelled high frequency characteristics [40].

6. Pi operator.

   By defining the bilinear transformation (BLT) as an operator, it is possible to introduce an operator which generalizes all of the above operators. We have proposed the *pi* operator [5] as

$$\pi = \frac{2}{\Delta} \frac{(c_1 q - c_2)}{(c_3 q + c_4)} \tag{15}$$

   with the restriction that $c_1 c_4 \neq c_2 c_3$ (to ensure π is not a constant function [50]). The bilinear mapping produced has a pole at $q = -c_4/c_3$. By appropriate setting of the c_1, c_2, c_3, c_4 parameters, the pi operator can be reduced to each of the previous operators. Independently, the same operator was proposed in [19] with all coefficients set to unity.

Written in terms of the backward shift operator q^{-1}, the backward ADTOs are defined as:

$$\delta^{-1} = \frac{\Delta q^{-1}}{1 - q^{-1}}$$

$$\delta_m^{-1} = \frac{\Delta q^{-1}}{1 - c q^{-1}}$$

$$\begin{aligned}
\gamma^{-1} &= \frac{cq^{-1}}{1-(1-c)q^{-1}} \\
\rho^{-1} &= \frac{c_2\Delta q^{-1}}{1-(1-c_1\Delta)q^{-1}} \\
\pi^{-1} &= \frac{\Delta}{2}\frac{c_3+c_4q^{-1}}{c_1-c_2q^{-1}}
\end{aligned} \tag{16}$$

In this paper, we will focus on the first order operators, i.e., we will not consider second order operators, e.g., the second order gamma operator.

3 Stability Regions of ADTOs

Stability can be considered in two respects: (a) stability of the whole model and (b) stability of the operator itself. The introduction of a different operator into a model, means that the stability region is changed from the usual unit circle in the complex plane, to some other region (ie the ν-plane). Secondly, the operators themselves have a recursive structure, and are therefore subject to stability constraints themselves. In this section, we will only consider the case of operator stability.

3.1 Delta Operator

Mansour, Kraus and Jury considered the problem of robust stability of a system described by a delta polynomial whose parameters may be perturbed in a prescribed interval [31]. The stability region of the delta operator is the interior of the circle given by $C : |q + 1/\Delta| = 1/\Delta$, i.e., centered at $(-1/\Delta, 0)$ with radius $1/\Delta$ [35].

3.2 Modified Delta Operator

The stability region for the modified delta operator is the interior of the circle $C : |q + c/\Delta| = c/\Delta$, i.e., centered at $(-c/\Delta, 0)$ with radius c/Δ. For the δ_m-operator to be stable and defined requires that

$$0 < c < 1$$

3.3 Gamma Operator

The stability region for the gamma operator model is the interior of the circle $C : |q + \frac{(1-c)}{c}| = \frac{1}{c}$, i.e., centered at $(-\frac{(1-c)}{c}, 0)$ with radius $\frac{1}{c}$. Stability is maintained by ensuring $0 < c < 2$.

3.4 Rho Operator

The stability region for a ρ operator model is the interior of the circle C: $\left|q + \frac{(1-c_1\Delta)}{c_2\Delta}\right| = \frac{c_1}{c_2} + \frac{(1-c_1\Delta)}{c_2\Delta}$ i.e., centered at $(-\frac{1-c_1\Delta}{c_2\Delta}, 0)$ with radius $\frac{1}{c_2\Delta}$.
For the ρ operator to be stable and defined requires that

$$\begin{aligned}
0 &< c_1\Delta < 2 \\
0 &< c_2
\end{aligned}$$

3.5 Pi Operator

The stability region for the pi operator is found by applying the bilinear transformation $\pi = \frac{2}{\Delta}\frac{(c_1q-c_2)}{(c_3q+c_4)}$ to the unit circle in the $q-$plane to find the resulting region in the $\pi-$plane. This region can be found by applying the bilinear transformation in the following stages [50]:

1. Linear transformation

$$\pi_1 \quad = \quad c_3 q + c_4$$

In this case, a unit disc in the $q-$plane is transformed to a disk of radius c_3, and centre $(c_4, 0)$ in the π_1-plane.

2. Inversion

$$\pi_2 \quad = \quad \frac{1}{\pi_1}$$

Depending on the location of the perimeter of the disk in the π_1-plane, the resulting transformation is either to a plane or the exterior of a disk in the π_2-plane. If the circle in the π_1-plane does not pass through the origin, (i.e., $c_3 \neq c_4$), the image will be another circle, with the equation:

$$\text{C: } \left| q + \frac{c_4}{c_3^2 - c_4^2} \right| \quad = \quad \frac{\sqrt{c_4^2 + 1}}{c_3^2 - c_4^2} \tag{17}$$

The stability region is the exterior of this circle. If the circle does pass through the origin, (i.e., $c_3 = c_4$), then the resulting image is a plane given by:

$$\text{P:} \frac{c_4}{c_3^2 - c_4^2} \quad = \quad -\frac{1}{2} \tag{18}$$

The stability region is the right-hand side of this plane.

3. Linear transformation

$$\pi \quad = \quad -\left(c_2 + \frac{c_1 c_4}{c_3} \right) \pi_2 + \frac{c_1}{c_3}$$

This final transformation is applied to either the circle or line obtained in the step above to obtain the final resulting region of stability. Thus, to determine the stability region for the pi operator requires considerably more computation and depends on the operator parameters.

For stability of the operator, $|c_2/c_1| < 1$, and also $c_1, c_3 > 0$.

Remarks.

1. The procedure for determining the stability regions in the $\pi-$operator model requires the stability region to be re-evaluated at each update instant.

2. A simpler approach may be adopted by fixing the c_i parameters. This avoids any difficulties in on-line adaptation of the coefficients, but is clearly more restrictive.

3. By restricting $c_3 = c_4$, the resulting stability region will always be a half-plane. This may offer the possibility of reducing the computational burden to determine model stability.

4. For the operator to be minimum phase, $|c_4/c_3| < 1$.

4 Linear Models

We will consider a generalisation of the usual linear models in this section using ADTOs. This will serve as a prelude to the study in later sections in applying ADTOs to neural networks.

4.1 MA(ν) Model

A model which generalizes the usual discrete-time linear moving average model, (i.e., a single layer network) is given by

$$\begin{aligned} \hat{y}(t) &= G(\nu,\theta)x(t) \qquad (19) \\ G(\nu,\theta) &= \sum_{i=0}^{M} b_i \nu^{-i} \\ \nu^{-i} &= \begin{cases} q^{-i} & \text{shift operator} \\ \delta^{-i} & \text{delta operator} \\ \delta_m^{-1} & \text{modified delta operator} \\ \gamma^{-i} & \text{gamma operator} \\ \rho^{-i} & \text{rho operator} \\ \pi^{-i} & \text{pi operator} \end{cases} \qquad (20) \end{aligned}$$

This general class of moving average model can be termed MA(ν).
We define $u_0(t) \triangleq x(t)$, and $u_i(t) \triangleq \nu^{-1}u_{i-1}(t)$ and hence obtain

$$u_i(t) = \begin{cases} x(t-i) & \text{shift operator} \\ \Delta u_{i-1}(t-1) + u_i(t-1) & \text{delta operator} \\ \Delta u_{i-1}(t-1) + c u_i(t-1) & \text{modified delta operator} \\ c u_{i-1}(t-1) + (1-c)u_i(t-1) & \text{gamma operator} \\ c_2 \Delta u_{i-1}(t-1) + (1 - c_1\Delta)u_i(t-1) & \text{rho operator} \\ & \text{bomega operator} \\ \frac{\Delta}{2c_1}\left(c_3 u_{i-1}(t) + c_4 u_{i-1}(t-1)\right) - \frac{c_2}{c_1}u_i(t-1) & \text{pi operator} \end{cases} \qquad (21)$$

4.2 ARMA(ν) Model

In the case of an autoregressive moving average ARMA(ν) model, we have

$$G(\nu,\theta) = \frac{\sum_{i=0}^{M} b_i \nu^{-i}}{1 + \sum_{i=1}^{N} a_i \nu^{-i}} \qquad (22)$$

Corresponding to (21), we may similarly obtain the equations describing the propagation of signals through each operator. Note that in the MA model case, the MA(q) model is FIR, while the MA(ν) model is IIR. On the other hand, each ARMA model is IIR. The equations describing the ARMA models can be obtained as in (21).

5 A Special Class of Nonlinear Models

Inspired by the models shown in section 4, it is possible to define new classes of MLP architectures which may be biased towards reduction of sensitivity. These models will be considered in this section.

5.1 MLP(ν) Model

A nonlinear model may be defined using a multilayer perceptron (MLP) with the ν-operator elements at the input stage. This model is termed the ν-operator multilayer perceptron or MLP(ν) model. An MLP(ν) model having L layers with $N_0, N_1, ..., N_L$ nodes per layer, is defined in the same manner as a usual MLP (see equation (2)), with

$$z_k^l(t) = f\left(\hat{x}_k^l(t)\right) \qquad (23)$$

$$\hat{x}_k^l(t) \quad = \quad \begin{cases} \sum_{i=1}^{N_l} w_{ki}^l z_i^{l-1}(t) & l = 2, ..., L \\ \sum_{i=1}^{N_1} w_{ki}^1 \nu^{-i} x(t) & l = 1 \end{cases} \tag{24}$$

The case we consider employs the ν-operator at the input layer only, however, it is also possible to introduce operators throughout the network as required (see for example [3]). The case where each synapse is replaced by gamma operators is considered in [28]. We will refer to the special case when $\nu = q$ as a MLP(q) model.
Note that this class of models is new, in that it is specifically designed to reduce the sensitivity of the MLP(q) models. It is a subclass of the IIR MLP architectures, except that the IIR part is formed in a very specific manner, as indicated. This architecture also implies that the IIR MLP can be further generalized to use alternative discrete-time operators in the IIR synapses.

5.2 Multiple pole models

Each operator as designated in (16) has just a single pole. An MLP(ν) model which has multiple operators, each with different poles, was proposed in [45]. This was termed a "focussed network". In this case, we have

$$z_k^l(t) \quad = \quad f\left(\hat{x}_k^l(t)\right) \tag{25}$$

$$\hat{x}_k^l(t) \quad = \quad \begin{cases} \sum_{j=1}^{N_l} w_{kj}^l z_j^{l-1}(t) & l = 2, ..., L \\ \sum_{j_1=1}^{N_1} w_{kj_1 1}^1 \nu_1^{-j_1} \cdots \sum_{j_M=1}^{N_1} w_{kj_M M}^1 \nu_M^{-j_M} x(t) & l = 1 \end{cases} \tag{26}$$

This approach can be applied to each of the architectures we consider in this paper.

5.3 Individual pole models

The models considered so far assume that for a cascade of operators, ie

$$z(t) \quad = \quad \sum_{i=1}^{M} \hat{w}_i \nu^{-i} x(t) \tag{27}$$

every operator in sequence has identical parameters. This approach follows the classical approach of the shift operator, however, it is possible for each operator to be differently parametrized, enabling a more general structure. Thus we have

$$z(t) \quad = \quad 1 + \sum_{i=1}^{M} \prod_{j=1}^{i} w_i \nu_j^{-1} x(t) \tag{28}$$

This extension complicates the notation significantly and can be viewed as a "diagonal" (reduced complexity) multiple pole model, where $\hat{w}_i = w_{ii}$ (see (26)). Hence we do not consider this approach further in this paper.

5.4 Nonlinear model stability

The stability of the nonlinear models which employ linear dynamic subsections is trivial. The sigmoid function provides a self-stabilization effect, bounding the output of nonlinear units and preventing any 'flow-on' of instabilities from upsetting the rest of the network [4]. In practice, we have observed that if a linear dynamic section in a lower layer goes unstable, then the model stability can be maintained by providing the constraints

$$\begin{array}{ccccc} L_1 & < & a_j, b_j & < & U_1 \\ L_2 & < & \Delta a_j, \Delta b_j & < & U_2 \\ L_3 & < & \hat{x}(t) & < & U_3 \end{array}$$

where $U_1, U_2, U_3, L_1, L_2, L_3$ are upper and lower bounds on the weights, weight changes, and synaptic outputs implemented directly on the model. Thus, the model stability remains essentially unchanged as a result of introducing the alternative discrete-time operators. It is desirable however, that efforts be made to ensure stability of the individual operators.

6 Parameter Sensitivity Analysis

Various authors have considered the issue of sensitivity to errors in the weights in feedforward neural networks [2, 9, 14, 26, 38, 39, 56, 63]. Typically, these analyses are based on probabilistic methods. Minai and Williams [34] considered the issue of performance changes of a network due to perturbations in the individual output response of units within the network. Von Lehman et. al. [29] showed that weight discretization in a feedforward network resulted in very poor learning and performance. They observed that networks with less than 300 levels of quantization did not converge, but that adding noise to the network significantly improved the probability of convergence. A contribution of this paper, is to examine the issue of sensitivity in MLPs when modelling nonlinear dynamical systems. In contrast to the previous probabilistic approaches, we consider the issue from the perspective of linear systems theory and indicate a mechanism for obtaining some understanding of the problem of high sensitivity within feedforward network architectures.
For linear systems, the basis for analysing sensitivity properties is to examine the poles and zeros of the model. How can poles and zeros be considered in nonlinear systems ?
It is widely regarded as a fact that the use of poles and zeros is inappropriate in nonlinear systems. In fact, this perception is incorrect, and a substantial amount of work has been done in extending the concept of transfer functions and hence poles and zeros, to nonlinear systems (see for example [7, 49, 51]). The basis of the approach considered here stems from the fact that while a general nonlinear system may be given as $y = F(\cdot)$ where the structure of $F(\cdot)$ is completely unknown, we in fact consider a special class of nonlinear system, for which it is possible to describe $F(\cdot)$ is some detail.
More specifically, consider a nonlinear model described by

$$y(t) \quad = \quad F\left(G(q)x(t)\right) \tag{29}$$

where $F(\cdot)$ is some memoryless nonlinearity and $G(q) = B(q)/A(q)$. Note that for $G(q) = B(q)$, (29) is equivalent to a TDNN. It is well known that such a model (29) can be approximated by a Volterra series [60].
It is desired to obtain a measure of the coefficient sensitivity for the model (29). In order to simplify the analysis presented here, we consider a simple TDNN model structure which has only one hidden unit and one layer. It is clear that this analysis can be extended to the general MLP(q) case. Hence we have

$$y(t) = f\left(b_1 x(t) + b_2 x(t-1) + ... + b_n x(t-n)\right) \tag{30}$$

where $f(u) = 1/(1+e^{-\alpha u})$ is a sigmoid activation function, α is the activation function parameter. Following the usual derivation of a Volterra series [49], the activation function $f(\cdot)$ can be approximated by a power series expansion [62]

$$f(u) = \sum_{j=0}^{\infty} \xi_j \, u^j \tag{31}$$

where for a finite pth-order expansion around $u = 0$, $\xi_0 = 1/2$, $\xi_1 = \alpha/4$, $\xi_3 = \alpha^3/48$, $xi_5 = \alpha^5/480$, $xi_7 = \frac{17}{80640}\alpha^7$, ... For sufficiently small u, the finite series expansion will be convergent.
Hence we can approximate (30) by

$$y(t) \quad = \quad \sum_{i=0}^{p} \xi_i \left(b_0 x(t) + b_2 x(t-1) + ... + b_n x(t-n)\right)^i + ...$$

$$\approx \sum_{i_0=0}^{p}\sum_{i_1=0}^{p}...\sum_{i_n=0}^{p}\beta(i_0,i_1,...,i_n)x^{i_0}(t)x^{i_1}(t-1)...x^{i_n}(t-n) \tag{32}$$

Eqn (32) defines the usual Volterra series expansion and $\{\beta\}$ are normally termed the Volterra kernels [51] (see also [62] for a discussion on some issues of extracting the Volterra kernels from MLPs). Note that for the activation function considered here, only odd $\{\xi\}$ terms are nonzero, however for other activation functions, and for the general MLP case, this will not necessarily be so. In contrast to the usual problem of estimating the Volterra kernels, here $\{\beta\}$ can be found directly from the MLP parameters $\{b\}$ and the activation function parameter α.
Expanding (32), we have

$$\begin{aligned} y(t) &= \sum_{k=0}^{n}\bar{\beta}_{0k}+\sum_{k=0}^{n}\bar{\beta}_{1k}x(t-k)+...+\sum_{k=0}^{n}\bar{\beta}_{pk}x^p(t-k) \\ &\quad +O\left(x^{i_0}(t)x^{i_1}(t-1)...x^{i_n}(t-n)\right)+... \end{aligned} \tag{33}$$

where

$$\begin{aligned} \bar{\beta}_{j0} &= \beta(j,0,...,0) \\ \bar{\beta}_{j1} &= \beta(0,j,...,0) \\ \vdots &\quad \vdots \\ \bar{\beta}_{jn} &= \beta(0,0,...,j) \qquad j=0,...,p \end{aligned} \tag{34}$$

Eqn (33) can be written as

$$\begin{aligned} y(t) &= \sum_{k=0}^{n}\bar{\beta}_{0k}+g_1(x(t))+...+g_p(x^p(t)) \\ &\quad +O\left(x^{i_0}(t)x^{i_1}(t-1)...x^{i_n}(t-n)\right)+... \end{aligned} \tag{35}$$

where $y(t)$ is comprised of terms which include

$$g_j(x^j(t)) = \Psi_j(q)x^j(t) \tag{36}$$

$$\Psi_j(q) = \sum_{k=0}^{n}\bar{\beta}_{jk}q^{-k} \tag{37}$$

Hence it is clear from (35)-(37) that $y(t)$ can be approximated by a summation of subsystems, some of which have linear transfer functions as in (37). For these transfer functions $\Psi_j(q)$, it is evident that the parameter sensitivity measure in (8) is applicable. This indicates that the problem of parameter sensitivity exists in the model (29) and hence, FIR MLP neural networks.
Moreover, $\{\Psi_j(q)\}$ can be replaced by $\{\Psi_j(\nu)\}$ which will result in lower parameter sensitivity in the transfer function, provided the appropriate parametrization of the alternative discrete-time operator ν is selected [19]. Given

$$\begin{aligned} \bar{A}(\nu) &= \sum_{i=0}^{n}\bar{a}_i\nu^{-i} \\ &= \nu^{-n}\prod_{i=1}^{n}(\nu-\bar{\lambda}_i) \end{aligned} \tag{38}$$

$$\nu = \frac{q-c_1}{c_2} \tag{39}$$

it can be shown that

$$\bar{S}_{ij} < S_{ij} \quad \text{for } c_2 < 1.0 \tag{40}$$

where $|c_1| < 1.0$ and

$$\begin{aligned} \bar{S}_{ij} &= \frac{d\bar{\lambda}_i}{d\bar{a}_j} \\ &= \frac{\bar{\lambda}_i^{n-j}}{\prod_{k \neq i}(\bar{\lambda}_i - \bar{\lambda}_k)} \end{aligned} \tag{41}$$

Thus $\Psi(\nu)$ will have a lower parameter sensitivity than $\Psi(q)$ as required.
Although the power series expansion has validity only over a certain range of input signals, the analysis shows the possible presence of parameter sensitivity in neural network models such as the TDNN and FIR MLP.
From this analysis we may make the following observations.

Observation 1. An MLP(q) model may be subject to problems of high parameter sensitivity.

Observation 2. A neural network MLP(ν) comprised of alternative discrete-time operators can have lower parameter sensitivity than an MLP(q) neural network comprised of the usual shift operators.

Remarks.

1. It is straightforward to observe that ADTOs can be used to improve the performance in FWL implementations.

2. The above analysis makes a critical assumption concerning the nonlinearity element, viz., it is smooth and continuous. Thus, the analysis cannot be applied to nonlinearities which are nonsmooth, e.g., a threshold nonlinearity. For the types of neural networks we are concerned with, this is not a problem.

7 Learning Algorithms for Alternative Operators

7.1 Derivation of algorithms

In contrast to shift operator models, the ADTOs require parametrization[3]. One method to do this is by the use of on-line learning algorithms.
On-line algorithms to update the operator parameters in the MA(ν) model can be found readily. A recursive least squares (RLS) algorithm was recently derived for delta operator ARMA filters by Fan and Li [16]. In the case of the MLP(ν) model, we approach the problem by backpropagating the error information to the input layer and using this to update the operator coefficients. De Vries and Principe et. al., proposed stochastic gradient descent type algorithms for adjusting the c operator coefficient using a least-squares error criterion [11, 44]. For brevity, we omit the updating procedures for the MLP network weights; a variety of methods may be applied (see for example [46, 52]), but instead concentrate on the equations for updating the parameters in the ADTOs.
In deriving a learning algorithm, a common approach is to use a cost criterion such as the instantaneous output error defined as $J(t) = \frac{1}{2}e^2(t)$, where $e(t) = y(t) - \hat{y}(t)$.

[3] In practice we have found that it is possible to simply fix the ν-operator parameters to values which provide reduced sensitivity as indicated in (40). For optimization of these parameters to some value which reduces the output error it may be useful to adapt the parameters on-line. Hence the algorithms in this section are presented.

As noted in introduction, an advantage of ADTOs, is that the sensitivity can be reduced as compared to using the shift operator. Is it possible to simultaneously minimize the output error and reduce sensitivity ?
Consider, for example, the ρ operator. The two constraints required to reduce sensitivity are

1. $|c_1| < 1.0$
2. $|c_2| < 1.0$

The first constraint is satisfied due to the stability requirement for the model. The second constraint can be enforced to ensure sensitivity is reduced. When an on-line learning algorithm is used, we propose that several approaches may be used. If sensitivity is not an issue, then constraint 2 can be ignored. If, on the other hand, we require that sensitivity be reduced, then constraint 2 can be enforced. Another approach is to introduce the constraint as a regularization term. In this case, we could use $J'(t) = \alpha\frac{1}{2}e^2(t) + (1-\alpha)c_2^p$ where $p \geq 2$ is a constant integer factor defining the curve of the regularization term with respect to c_2; α is a constant such that $0 \leq \alpha \leq 1$. The learning algorithms are considered below.
Defining $\hat{\theta}(t)$ as the estimated operator parameter vector at time t of the parameter vector θ, we have

$$\hat{\theta} = \begin{cases} \hat{c} & \text{gamma operator} \\ [\hat{c}_1, \hat{c}_2]' & \text{rho operator} \\ [\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4]' & \text{pi operator} \end{cases} \tag{42}$$

The parameter in the modified delta operator is fixed and hence we do not consider a learning algorithm in this case.
A first order algorithm to update the coefficients is

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \Delta\hat{\theta}(t) \tag{43}$$

$$\tag{44}$$

$$\Delta\hat{\theta}(t) = -\eta\nabla_\theta J(\theta;t) \tag{45}$$

and the adjustment in weights is found as

$$\begin{aligned} \Delta\hat{\theta}(t) &= -\eta\frac{\partial J(t)}{\partial\theta} \\ &= \eta\sum_{i=1}^{M}\delta_i^1(t)\psi_i(t) \end{aligned} \tag{46}$$

where $\delta_i^1(t)$ is the backpropagated error at the ith node of input layer, and $\psi_i(t)$ is the first order sensitivity vector of the model operator parameters, defined respectively by

$$\begin{aligned} \delta_k^l(t) &= -\frac{\partial J(t)}{\partial\hat{x}_k^l(t)} \\ &= e_k(t)f'(\hat{x}_k^l(t)) && l = L \\ &= f'\left(\hat{x}_k^l(t)\right)\sum_{p=1}^{N_l+1}\delta_p^{l+1}(t)w_{kp}^{l+1} && 1 \leq l \leq L-1 \end{aligned} \tag{47}$$

$$\psi_i(t) = \begin{cases} \frac{\partial u_i(t)}{\partial c} & \text{gamma operator} \\ \left[\frac{\partial u_i(t)}{\partial c_1}, \frac{\partial u_i(t)}{\partial c_2}\right]' & \text{rho operator} \\ \left[\frac{\partial u_i(t)}{\partial c_1}, \frac{\partial u_i(t)}{\partial c_2}, \frac{\partial u_i(t)}{\partial c_3}, \frac{\partial u_i(t)}{\partial c_4}\right]' & \text{pi operator} \end{cases} \tag{48}$$

If the sensitivity regularization term is included, then we replace (46) with

$$\begin{aligned}\Delta\hat{\theta}(t) &= -\eta\frac{\partial J'(t)}{\partial\theta} \\ &= \begin{cases} \eta\left[\alpha\sum_{i=1}^{M}\delta_i^1(t)\psi_i(t)+(1-\alpha)\left(pc_2^{p-1}\right)\right] & \theta_j = c_2 \forall j \\ \eta\left[\alpha\sum_{i=1}^{M}\delta_i^1(t)\psi_i(t)\right] & \theta_j \neq c_2 \forall j \end{cases}\end{aligned} \tag{49}$$

Hence the regularization term only affects c_2 directly and can be included in the algorithm implementation if required.
Substituting $u_i(t)$ in from (21), the recursive equations for $\psi_i(t)$ are

$$\psi_i(t) = u_{i-1}(t-1) - u_i(t-1) + \hat{c}_i\psi_{i-1}(t-1) + (1-\hat{c})\psi_i(t-1) \quad \text{gamma operator}$$

$$\psi_i(t) = \begin{bmatrix} \hat{c}_2\Delta\psi_{i-1,1}(t-1) + (1-\hat{c}_1\Delta)\psi_{i,1}(t-1) - \Delta u_i(t-1) \\ \Delta u_{i-1}(t-1) + \hat{c}_2\Delta\psi_{i-1,2}(t-1) + (1-\hat{c}_1\Delta)\psi_{i,2}(t-1) \end{bmatrix} \quad \text{rho operator}$$

$$\psi_i(t) = \begin{bmatrix} \frac{\Delta}{2\hat{c}_1}\left(\hat{c}_3\psi_{i-1,1}(t) + \hat{c}_4\psi_{i-1,1}(t-1)\right) + \frac{\hat{c}_2}{\hat{c}_1}\psi_{i,1}(t-1) \\ -\frac{\Delta}{2\hat{c}_1^2}\left(\hat{c}_3 u_{i-1}(t) + \hat{c}_4 u_{i-1}(t-1)\right) - \frac{\hat{c}_2}{\hat{c}_1^2}u_i(t-1), \\ \frac{\Delta}{2\hat{c}_1}\left(\hat{c}_3\psi_{i-1,2}(t) + \hat{c}_4\psi_{i-1,2}(t-1)\right) + \frac{\hat{c}_2}{\hat{c}_1}\psi_{i,2}(t-1) + \frac{1}{\hat{c}_1}u_i(t-1), \\ \frac{\Delta}{2\hat{c}_1}\left(u_{i-1}(t) + \hat{c}_3\psi_{i-1,3}(t) + \hat{c}_4\psi_{i-1,3}(t-1)\right) + \frac{\hat{c}_2}{\hat{c}_1}\psi_{i,3}(t-1), \\ \frac{\Delta}{2\hat{c}_1}\left(\hat{c}_3\psi_{i-1,4}(t) + u_{i-1}(t-1) + \hat{c}_4\psi_{i-1,4}(t-1)\right) + \frac{\hat{c}_2}{\hat{c}_1}\psi_{i,4}(t-1) \end{bmatrix} \quad \text{pi operator}$$

for the gamma, rho, and pi operators respectively, and where $\psi_{i,j}(t)$ refers to the jth element of the ith ψ vector, with $\psi_{i,0}(t) = 0$.
A more powerful updating procedure can be obtained by using the Gauss-Newton method [30]. In this case, we replace (46) with (omitting i subscripts for clarity),

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \gamma(t)R^{-1}(t)\psi(t)\Lambda^{-1}\delta(t) \tag{50}$$

where $\gamma(t)$ is the gain sequence (see [30] for details), Λ^{-1} is a weighting matrix which may be replaced by the identity matrix [54], or estimated as [30]

$$\hat{\Lambda}(t) = \hat{\Lambda}(t-1) + \gamma(t)\left(\delta^2(t) - \hat{\Lambda}(t-1)\right) \tag{51}$$

$R(t)$ is an approximate Hessian matrix, defined by

$$R(t+1) = \lambda(t)R(t) + \zeta(t)\psi(t)\psi'(t) \tag{52}$$

where $\lambda(t) = 1 - \zeta(t)$. Efficient computation of R^{-1} may be performed using the matrix inversion lemma [55], factorization methods such as Cholesky decomposition or other fast algorithms.
Using the well known matrix inversion lemma [30], we substitute $P(t) = R^{-1}(t)$, where

$$P(t) = \frac{1}{\lambda(t)}P(t) - \frac{\zeta(t)}{\lambda(t)}\left(\frac{P(t)\psi(t)\psi'(t)P(t)}{\lambda(t) + \zeta(t)\psi'(t)P(t)\psi(t)}\right) \tag{53}$$

The initial values of the coefficients are important in determining convergence. Principe et. al. [44] have shown that the error surface may be multimodal. For the gamma operator, setting the coefficients to unity can provide the best approach for certain problems. This is discussed further in the next section.

7.2 Convergence of operator parameters

An essential task of any learning algorithm is to ensure that the parameters converge to some suitable values to allow the model to perform capably. In this section, we discuss some aspects of the convergence of the operator learning algorithms proposed in Section 7.

7.2.1 Modality of mean square output error surface

The ADTOs discussed in this paper are all IIR filters [47]. It is known however, that for IIR filters, the mean square output error (MSOE) surface can be multimodal with respect to parameters, [24]. That is, there may exist more than one minimum on the surface. This poses a difficulty for gradient descent algorithms which may become lodged in a local minimum which has a solution significantly worse than an optimal global minimum point.

It has been shown that the problem of multimodality in linear IIR models lies largely with the difference between the order of the system and the order of the model [17, 48]. In particular, reduced order models are known to contribute to the problem of multimodality [24]. For more complex models such as those we consider here, where there are a number of repeated IIR type operators, the situation is not so obvious. One way in which the situation can be handled, is to examine the error surface properties of a direct-form IIR model which is equivalent to the alternative operator model.

From this viewpoint, one can quickly see that there may indeed be situations where the model has reduced order with respect to the system. Hence we would expect that a multimodal error surface may result in terms of the operator parameters. Indeed, Principe et. al. have shown this to be the case for the gamma operator [44] and it is simple to show that this would also be the same for slightly more complex operators. What can be done about this problem ?

There are a number of possible solutions which we propose here. Since the main purpose in this paper is to raise the issue of ADTOs in the context of neural network signal processing, we do not attempt to consider these approaches in detail. Indeed, the problem is a significant open problem in linear systems theory [23, 53, 55] and we do not seek to solve the problem in this paper. Rather we present these ideas as possible avenues for future research and as a means of overcoming the difficulties encountered.

A key point in the issues raised in the following sections stems from this basic concept:

> *The operator parameters can be regarded as any other parameter within the model, or they can be treated in some different manner.*

7.2.2 Alternative cost functions for operator parametrization

Due to the problems of multimodality in the MSOE surface of IIR systems, is it possible to use some other cost function to parametrize the operators ?

An alternative criterion for determining the operator parameters may be derived by considering the numerical conditioning of the data covariance matrix due to the operators. In order to be able to use this criterion however, we need to consider the properties of the models. If a model is chosen such that some other cost function is minimized, what will happen to the output error ?

Consider the fundamental representational properties of the linear subsystems.

Suppose we have ARMA models $H(q) = B(q)/A(q)$, $H'(\nu) = B'(\nu)/A'(\nu)$. Since the order of the operator numerator and denominator is the same in each case, it is clear that no extra modelling capability is provided by the alternative operator model. Therefore, *it does not matter what values of ν operator parameters we pick*, it will always be possible to find a set of $\{a'_i, b'_i\}$ parameters which give the same filter transfer function as $H(q)$! Since it does not matter what operator parameters are chosen, it is possible to select operator parameters based on any method we choose, not necessarily based on minimization of the output error.

Hence, for ARMA(ν) models and models with ARMA(ν) inputs, we draw the conclusion that it is not necessary to use the MSOE as a criterion for parametrizing the operator parameters. Importantly, this opens up the possibility of overcoming the problems of local minima in the MSOE surface by using a completely different optimization criteria. It is also possible to introduce the other criteria as regularizers.
Consider now an MA model $H_f(q) = B(q)$. The above approach is not necessarily applicable to MA model structures. This is due to the fact that a given MA(q) model cannot be transformed to a corresponding ν-operator representation which with MA(ν) structure, ie $H'_f(\nu) = B'(\nu)$. That is, it is not necessarily possible to find a set of $\{b'_i\}$ parameters which give the same filter transfer function as $H_f(q)$. Therefore it cannot be guaranteed that by choosing any particular operator parameters, we would be able to then minimize the output error by manipulation of the $\{b'_i\}$ parameters[4].
However, by appropriate choice of operator structure based on knowledge of the system structure, it may be possible to ensure the model has a unique global minimum which would overcome the problem.

7.2.3 Numerical conditioning as a cost function

As indicated above, it is possible to select operator values which improve the numerical conditioning of the data covariance matrix [6, 19]. It was shown in [6] that significant improvements can be obtained in the conditioning of the data covariance matrix by appropriate selection of the operator parameters. This provides a means of improving the convergence of the rest of the model. For discussions on this topic using the delta operator and a special case of the Pi operator, see [19].

8 Experimental results

In this section, we provide some basic results which indicate the possible improvements in performance that can be obtained through the use of the proposed alternative discrete-time operators.
We are primarily interested in the differences between the operators themselves for modelling and prediction, and not the associated difficulties of training multilayer perceptrons (recall that our models will only differ at the input layer). For the purposes of a more direct comparison, in this paper we test the models using a single layer network. Hence these linear system examples are used to provide an indication of the operators' performance. In these experiments we did not seek to minimize the sensitivity by using the regularization term.

Experiment 1

The first problem considered is a system identification task arising in the context of high bit rate echo cancellation [16]. In this case, the system is described by

$$H(z) = \frac{0.0254 - 0.0296q^{-1} + 0.00425q^{-2}}{1 - 1.957q^{-1} + 0.957q^{-2}} \tag{54}$$

This system has poles on the real axis at 0.9994, and 0.9577, thus it is an LDLF system. The input signal to the system in each case consisted of uniform white noise with unit variance. A Gauss-Newton algorithm was used to determine all unknown weights. We conducted Monte-Carlo tests using 20 runs of differently seeded training samples each of 2000 points to obtain the results reported. We assessed the performance of the models by using the Signal-to-Noise Ratio (SNR) defined as $10\log(E[d(t)^2]/E[e(t)^2])$, where $E[\cdot]$ is the expectation operator, and $d(t)$ is the desired signal. For each run, we used the last 500 samples to compute a SNR figure.

[4] If it could be established that a given MA(ν) model has the same global minimum as an MA(q) model, then it would be possible to freely use the MA(ν) model, selecting the ν operator parameters according to whatever criterion is desired.

Table 1: System Identification Experiment 1 Results

Model Operator	Avg SNR (dB)	Best SNR (dB)
shift	+2.7	+3.6
delta	-7.1	+7.7
gamma	+5.7	+14.1
rho	+9.7	+16.5
pi	+10.0	+16.5

Table 2: System Identification Experiment 2 Results

Model Operator	Avg SNR (dB)	Best SNR (dB)
shift	+10.7	+12.3
delta	-21.5	+10.2
gamma	+13.5	+15.0
rho	+13.3	+17.4
pi	+14.0	+17.9

For the purposes of this experiment, we conducted several trials and selected $\theta(0)$ values which provided stable convergence. The values chosen for this experiment were: $\theta(0) = \{0.75, [0.5, 0.75], [0.75, 0.7, 0.35, -0.25]\}$ for the gamma, rho and pi operator models respectively. In each case we used model order $M = 8$.
Results for this experiment are shown in Table 1 and Figure 1. We observe that the pi operator gives the best performance overall. Some difficulties with instability occurring were encountered, thereby requiring a stability correction mechanism to be used on the operator updates. The next best performance was observed in the rho and then gamma models, with fewer instability problems occurring.

Experiment 2

The second experiment used a model described by

$$H(z) = \frac{1 - 0.8731q^{-1} - 0.8731q^{-2} + q^{-3}}{1 - 2.8653q^{-1} + 2.7505q^{-2} - 0.8843q^{-3}} \tag{55}$$

This system is a 3rd order lowpass filter tested in [45]. The same experimental procedures as used in Experiment 6.1 were followed in this case.
For the second experiment (see Table 2), it was found that the pi operator gave the best results recorded over all the tests. On average however, the improvement for this identification problem is less. It is observed that the pi model is only slightly better than the gamma and rho models. Interestingly, the gamma and rho models had no problems with stability, while the pi model still suffered from convergence problems due to instability. As before, the delta model gave a wide variation in results and performed poorly.
The rho model was able to perform better than the gamma model on the problems tested, and gave similar results in terms of susceptibility to convergence and instability problems. The pi model appears capable of giving the best performance overall, but requires more attention to ensure the stability of the coefficients.
From these and other experiments performed it appears that performance advantages can be obtained through the use of the more complex operators. As observed from the best recorded runs,

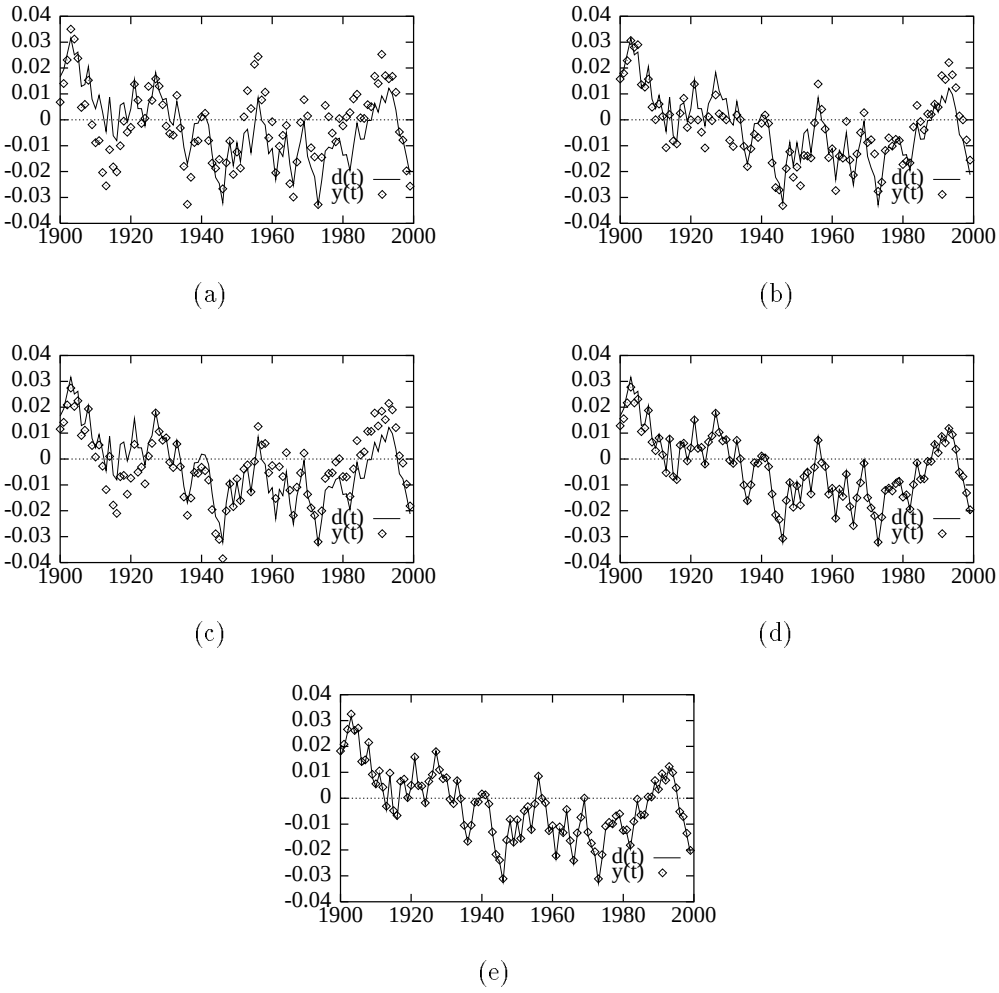

Figure 1: Comparison of typical model output results for Experiment 1 with models based on the following operators: (a) shift, (b) delta (c) gamma, (d) rho, and (e) pi.

the extra degrees of freedom in the rho and pi operators appear to provide the means to give better performance than the gamma model. The improvements of the more complex operators come at the expense of potential convergence problems due to instabilities occurring in the operators and a potentially multimodal mean square output error surface in the operator parameter space.

There is a problem with this approach however, in that while we have observed good results using the gradient descent algorithm to adjust the operator parameters, it is known that the mean square error surface of IIR filters may be multimodal [24, 44]. Hence there is no guarantee that a gradient descent algorithm will converge to the optimal solution.

There are different ways to view this situation. One way is to treat it as a significant problem and seek to find ways to obtain the global minimum. Another approach is to consider that it is not necessary to adjust the ADTO parameters to find the best *mean square output error*, but rather, treat the operators as *data preprocessors* within the main model which is itself adjusted to minimize the MSOE. In this way, it is not essential to use the MSOE as a criterion of adjustment, but we may consider other criteria as discussed in the previous section.

The results obtained indicate that the more complex operators provide a potentially more powerful modelling structure, with significant improvements over shift operator models. Clearly, there is a need for further investigation into the performance of these models on a wider range of tasks. We present these preliminary examples as an indication of how these alternative operators perform on some system identification problems.

9 Conclusions

Current neural network models frequently use the shift operator as a means of introducing time-dependence into the network structure. In linear systems theory, it is known that there may be problems associated with the conventional shift operator, and so the delta and rho operators have been previously introduced with consequent advantages in terms of sensitivity and model performance. To the best of our knowledge, these have not been previously considered in neural network models. However, we have shown that a recently introduced neural architecture, the Gamma model, is in fact a generalization of these operators introduced in the context of adaptive control and linear systems. Hence it is evident that the Gamma model will have performance advantages in finite word length models in terms of modelling accuracy and sensitivity, in addition to the architectural advantages described by de Vries and Principe in [12, 45, 44].

In this paper, models based on the delta operator, rho operator, and pi operator have been presented which can provide modelling advantages over models based on the shift operator in terms of sensitivity and hence improved robustness properties for finite word length implementations. Learning algorithms for the operators were derived.

With minor constraints, it is possible to ensure that models with alternative discrete-time operators have lower sensitivity than shift operator models while minimizing the mean square output error. The problem of local minima in the mean square output error surface remains largely unresolved at the time of writing, however we have proposed some possible areas for future work in this area.

It is apparent that models using alternative discrete-time operators offer a useful approach to provide improved numerical sensitivity and accuracy over conventional shift operator structures.

Acknowledgements

The first author acknowledges support from the Australian Research Council and the Frontier Research Program, RIKEN, Japan. The second author acknowledges partial support from the Australian Research Council.

References

[1] R.C. Agarwal and C.S. Burrus. New recursive digital filter structures having very low sensitivity and roundoff noise. *IEEE Trans. Circuits, Syst.*, CAS-22(12):921–927, 1975.

[2] C. Alippi, V. Piuri, and M. Sami. Sensitivity to errors in artificial neural networks: a behavioural approach. *IEEE Trans. Circuits, Syst. I: Fundamental Theory and Applications*, 42(6):358–361, 1995.

[3] A.D. Back and A.C. Tsoi. FIR and IIR synapses, a new neural network architecture for time series modelling. *Neural Computation*, 3(3):375–385, 1991.

[4] A.D. Back and A.C. Tsoi. Stabilisation properties of multilayer feedforward networks with time-delay synapses. In I. Aleksander and J. Taylor, editors, *Artificial Neural Networks 1*, volume 2, pages 1113–1116, Helsinki, 1992. Elsevier Science Publishers B.V. (North Holland).

[5] A.D. Back and A.C. Tsoi. A comparison of discrete-time operator models for nonlinear system identification. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 883–890, Cambridge, MA, 1995. The MIT Press.

[6] A.D. Back and A.C. Tsoi. Constrained pole-zero filters as discrete-time operators for system approximation. In F. Girosi, J. Makhoul, E. Manolakos, and E. Wilson, editors, *Proc. of the 1995 IEEE Workshop Neural Networks for Signal Processing 5 (NNSP95)*, pages 191–200, New York, NY, 1995. IEEE Press.

[7] S.P. Banks. *Mathematical Theories of Nonlinear Systems*. Prentice-Hall, London, 1988.

[8] Y. Bengio, R. de Mori, and M. Gori. Learning the dynamic nature of speech with backpropagation for sequences. *Pattern Recogition Letters*, 13:375–385, 1992.

[9] J.Y. Choi and C-H. Choi. Sensitivity analysis of multilayer perceptrons with differentiable activation functions. *IEEE Trans. Neural Networks*, 3:101–107, 1992.

[10] T.O. de Silva, P.G. de Oliveira, J.C. Principe, and B. de Vries. Generalized feedforward filters with complex poles. In S.Y. Kung, F. Fallside, J. Aa. Sorenson, and C.A. Kamm, editors, *Proc. of the 1992 IEEE Workshop Neural Networks for Signal Processing 2 (NNSP92)*, pages 503–510, Piscataway, NJ, 1992. IEEE Press.

[11] B. de Vries and J.C. Principe. A theory for neural networks with time delays. In R.P. Lippman, J.E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems*, volume 3, pages 162–168, San Mateo, CA, 1991. Morgan Kaufmann.

[12] B. de Vries and J.C. Principe. The Gamma model – a new neural model for temporal processing. *Neural Networks*, 5(4):565–576, 1992.

[13] B. de Vries, J.C. Principe, and P.G. de Oliveira. Adaline with adaptive recursive memory. In B.H. Juang, S.Y. Kung, and C.A. Kamm, editors, *Proc. of the 1991 IEEE Workshop Neural Networks for Signal Processing 1 (NNSP91)*, pages 101–110, New York, NY, 1991. IEEE Press.

[14] G. Dündar and K. Rose. The effects of quantization on multilayer perceptrons. *IEEE Trans. Neural Networks*, 6(6):1446–1451, 1995.

[15] J.L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[16] H. Fan and Q. Li. A δ-operator recursive gradient algorithm for adaptive signal processing. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Proc.*, volume III, pages 492–495. IEEE Press, 1993.

[17] H. Fan and M. Nayeri. On error surfaces of sufficient order adaptive IIR filters: proofs and counterexamples to a unimodality conjecture. *IEEE Trans. Acoust., Speech, Signal Processing*, 37(9):1436, 1989.

[18] P. Frasconi, M. Gori, and G. Soda. Local feedback multilayered networks. *Neural Computation*, 4:120–130, 1992.

[19] M. Gevers and G. Li. *Parameterizations in control, estimation and filtering problems: accuracy aspects*. Springer-Verlag, London, 1993.

[20] C.L. Giles, G.M. Kuhn, and R.J. Williams. Dynamic recurrent neural networks: Theory and applications. *IEEE Transactions on Neural Networks*, 5(2), 1994. Special Issue.

[21] G.C. Goodwin and K.S. Sin. *Adaptive Filtering, Prediction and Control*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

[22] J. Hertz, A. Krogh, and R. Palmer. *Introduction to Neural Computation.* Prentice-Hall, Englewood Cliffs, NJ, 1992.

[23] C.R. Johnson. Adaptive IIR filtering: Current results and open issues. *IEEE Trans. Inform. Theory*, 30(2):237–250, 1984.

[24] C.R. Johnson and M.G. Larimore. Comments on and additions to 'an adaptive recursive filter'. *Proc. IEEE*, 65(9):1399–1402, 1977.

[25] M.I. Jordan. Supervised learning and systems with excess degrees of freedom. Technical Report 88-27, Massachusetts Institute of Technology, COINS, 1988.

[26] P. Kerlirzin and P. Réfrégier. Theoretical investigation of the robustness of multilayer perceptrons: analysis of the linear case and extension to nonlinear networks. *IEEE Trans. Neural Networks*, 6:560–571, 1995.

[27] K. Lang, A.H. Waibel, and G.E. Hinton. A time delay neural network architecture for isolated word recognition. *Neural Networks*, 3:23–44, 1990.

[28] S. Lawrence, A.C. Tsoi, and A.D. Back. The gamma MLP for speech phoneme recognition. In D.S. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, Cambridge, MA, 1996. The MIT Press.

[29] A. Von Lehman, E.G. Paek, P.F. Liao, A. Marrakchi, and J.S. Patel. Influence of interconnection weight discretization and noise in an optoelectronic neural network. *Optics Letters*, 14:928–930, 1989.

[30] L. Ljung and T. Soderstrom. *Theory and Practice of Recursive Identification.* The MIT Press, Cambridge, MA, 1983.

[31] M. Mansour, F.J. Kraus, and E.I. Jury. On robust stability of discrete-time systems using delta-operators. In *Proc. American Control Conference*, pages 1417–1418. IEEE Press, 1992.

[32] P.E. Mantey. Eigenvalue sensitivity and state-variable selection. *IEEE Trans. Automat. Control*, AC-13(3):263–269, 1968.

[33] R.H. Middleton and G.C. Goodwin. *Digital Control and Estimation.* Prentice-Hall, Englewood Cliffs, NJ, 1990.

[34] A.A. Minai and R.D. Williams. Perturbation response in feedforward networks. *Neural Networks*, 7(5):783–796, 1994.

[35] T. Mori and I. Troch. Delta domain lyapunov matrix equation - a link between continuous and discrete equations. *IEICE Trans. Fundamentals*, E75-A:451–454, 1992.

[36] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1:4–27, March 1990.

[37] K. S. Narendra and K. Parthasarathy. Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Transactions on Neural Networks*, 2:252–262, March 1991.

[38] S.-H. Oh and Y. Lee. Sensitivity analysis of single hidden-layer neural networks with threshold functions. *IEEE Trans. Neural Networks*, 6:1005–1007, 1995.

[39] N.S. Orzechowski, S.R.T. Kumara, and C.R. Das. Performance of multilayer neural networks in binary-to-binary mappings under weight errors. In *Proc. ICNN93, San Francisco*, pages 1684–1689. IEEE Press, 1993.

[40] M. Palaniswami. A new discrete-time operator for digital estimation and control. Technical Report No. 1, The University of Melbourne, Department of Electrical Engineering, 1989.

[41] M. Palaniswami. Digital estimation and control with a new discrete time operator. In *Proc. 30th IEEE Conf. Decision and Control*, pages 1631–1632, New York, NY, 1991. IEEE Press.

[42] V. Peterka. Control of uncertain processes: Applied theory and algorithms. *Kybernetika*, 22:1–102, 1986.

[43] P. Poddar and K. P. Unnikrishnan. Non–linear prediction of speech signals using memory neuron networks. In B. H. Juang, S. Y. Kung, and C. A. Kamm, editors, *Neural Networks for Signal Processing: Proceedings of the 1991 IEEE Workshop*, pages 1–10. IEEE Press, 1991.

[44] J.C. Principe, B. de Vries, and P. Guedes de Oliveira. The Gamma filter - a new class of adaptive IIR filters with restricted feedback. *IEEE Trans. Signal Processing*, 41:649–656, 1993.

[45] J.C. Principe, B. de Vries, J.M. Kuo, and P. Guedes de Oliveira. Modeling applications with the focused gamma net. In J.E. Moody, S.J. Hanson, and R.P. Lippman, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 143–150, San Mateo, CA, 1992. Morgan Kaufmann.

[46] G.V. Puskorius and L.A. Feldkamp. Decoupled extended kalman filter training of feedforward layered networks. In *Proc. Int Joint Conf. Neural Networks*, volume I, pages 771–777, Seattle, 1991.

[47] L.R. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1975.

[48] P.A. Regalia. *Adaptive IIR Filtering in Signal Processing and Control*. Marcel-Dekker, New York, NY, 1995.

[49] W.J. Rugh. *Nonlinear Theory: The Volterra-Wiener Approach*. The Johns Hopkins University Press, Baltimore, Maryland, 1981.

[50] E.B. Saff and A.D. Snider. *Fundamentals of Complex Analysis for Mathematics, Science and Engineering*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[51] M. Schetzen. *The Volterra and Wiener Theory of Nonlinear Systems*. John Wiley and Sons, New York, NY, 1980.

[52] S. Shah and F. Palmieri. Meka - a fast local algorithm for training feedfoward neural networks. In *Proc. Int Joint Conf. Neural Networks*, volume III, pages 41–46, 1990.

[53] J.J. Shynk. Adaptive IIR filtering. *IEEE ASSP Magazine*, April:4–21, 1989.

[54] J.J. Shynk. Adaptive IIR filtering using parallel-form realizations. *IEEE Trans. Acoust., Speech, Signal Processing*, 37:519–533, 1989.

[55] T. Soderstrom and P. Stoica. *System Identification*. Prentice-Hall, London, 1989.

[56] M. Stevenson, R. Winter, and B. Widrow. Sensitivity analysis of feedforward neural networks to weight errors. *IEEE Trans. Neural Networks*, 1:71–90, 1990.

[57] A.C. Tsoi and A.D. Back. Locally recurrent globally feedforward networks, a critical review of architectures. *IEEE Trans. Neural Networks*, 5:229–239, 1994.

[58] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phonemic recognition using time delay neural networks. *IEEE Trans. Acoust., Speech, Signal Processing*, 37(3):328–339, 1989.

[59] E.A. Wan. Time series prediction by using a connectionist network with internal delay lines. In A.S. Weigend and N.A. Gershenfeld, editors, *Time Series Prediction: Forecasting the Future and Understanding the Past*, volume Proc. Volume XV Santa Fe Institute Studies in the Sciences of Complexity, pages 195–217, Reading,MA, 1994. Addison-Wesley.

[60] N. Wiener. *Nonlinear Problems in Random Theory*. The MIT Press, Cambridge, MA, 1958.

[61] D. Williamson. Delay replacement in direct form structures. *IEEE Trans. Acoust., Speech, Signal Processing*, 34(4):453–460, April 1988.

[62] J. Wray and G.G.R. Green. Calculation of the Volterra kernels of non-linear dynamic sytems using an artificial neural network. *Biological Cybernetics*, 71:187–195, 1994.

[63] Y. Xie and M.A. Jabri. Analysis of the effects of quantization in multilayer neural networks using a statistical model. *IEEE Trans. Neural Networks*, 3:334–338, 1992.