A neural network perspective to extended Luenberger observers

Deniz Erdogmus[1], **A. Umut Genç**[2], **José C. Principe**[1]
1 CNEL, Dept. of Electrical and Computer Engineering, University of Florida
2 Dept. of Engineering, University of Cambridge

Abstract

In this paper we investigate the use of adaptive extended Luenberger state estimators for general nonlinear and possibly time-varying systems. We identify the connection between the extended Luenberger observer and Grossberg's additive model for dynamic neural networks. The association between dynamic neural networks and the Luenberger observer leads to an obvious modification on the proposed observer scheme that would allow handling state estimation for those systems whose dynamic equations are partially known or not known at all. The performance of the adaptive observer is demonstrated on a number of systems including an LTI system, the Van der Pol oscillator, the Lorenz attractor and a realistic partial gasoline engine model.

Introduction

The potential of powerful mapping and representational capabilities of artificial neural network architectures has long been recognised in the neural network community[1]. The introduction of a backpropagation algorithm by Rumelhart *et al* enabled real-time applications of neural networks as adaptive systems[2]. Neural networks (NN) are being recognised in other fields of research including signal processing, communications, and control as valuable tools that offer simple solutions to difficult problems[3-5]. Especially in the area of adaptive controls, neural networks have experienced an increased interest in the last decade, due to their inherent adaptability and universal approximation properties. This interest was mainly ignited by the early works of Werbos, Shoureshi *et al*, Narendra and Parthasarathy, Gupta and Rao, and Miller *et al*.[6-10]. Although these first attempts were mostly heuristic in nature, there has been a stream of publications inspired by the idea, involving deeper analyses[11-14]. These later works have mostly focused on the application of recurrent neural networks for system identification and observer design, as well as adaptive and robust controllers for general nonlinear systems.

Puskorius and Feldkamp[15] investigate the application of recurrent multilayer perceptrons (MLP) to the control of nonlinear dynamical systems and propose an alternative training algorithm to update the dynamic weights of the network based on parameter-based extended Kalman filter (EKF) estimates. Their simulation results with a number of nonlinear systems favour the use of EKF-based training algorithm over the conventional backpropagation. Zhu *et al* focus on the application of dynamic recurrent neural networks (DRNN) as observers for nonlinear systems[11]. They consider a class of single-input-single-output (SISO) nonlinear time-varying systems in their work, where they prove the boundedness of the observer error and the DRNN weights during adaptation using Lyapunov stability theory and the well-known universal approximation theorem for neural networks[1,11]. With an alternative approach, Wang and Wu exploit the multilayer recurrent neural networks as matrix equation solvers and utilise this scheme to synthesise linear state observers in real-time by solving the Sylvester's equation for pole placement[16].

There are also examples of static feedforward neural network applications to observer and controller design. Ahmed and Riyaz consider an off-line training scheme for a MLP based observer design for nonlinear systems. They note that although the NN observer requires more computation in the training phase, it is more computation-efficient compared to the EKF in the implementation phase [17].

An interesting approach is presented in[13] by Vargas and Hemerly, where they employ linearly parametrized neural networks (LPNN) for the design of an adaptive observer for general nonlinear systems. LPNN include a wide class of networks including radial-basis-function (RBF) networks, adaptive fuzzy systems (with specific choices of rules and membership functions it can be shown that fuzzy systems are equivalent to RBF networks), and wavelet networks. They use Lyapunov stability theory to prove the stability of the observer and the neural network weights and demonstrate the performance of the designed observer on a single-link robot manipulator model. Fretheim *et al* on the other hand, utilise the feedforward MLP in the observer design problem with a little twist. They formulate the problem as a multi-step prediction, and exploit the extrapolation capabilities of the MLP to obtain the state estimates[18]. Ge *et al*, in their contribution to neurocontrol, use RBF networks in an adaptive output feedback NN controller scheme, for which they provide proofs and simulation studies for bounded tracking errors given sufficiently large networks[12]. Another theoretical line of study, directly related to the subject is the neural network approach to obtain approximate realisations of an unknown dynamic system solely from its input-output history. A recent result on this has been provided in[19]. In their work, Hovakimyan *et al* utilise a feedforward MLP in the prediction-framework to obtain a model of the unknown nonlinear dynamical system.

Besides all the theoretical approaches to the subject matter, there are also a number of application-oriented studies on the use of neural networks in controls. Most commonly investigated applications are observer and controller designs for robot manipulators[14-20], induction motors[21-23], synchronous motors[24,25] and finally air-fuel ratio (AFR) in gasoline engines[26,27,28]. In fact, the AFR problem has become more important recently as the need for more eco-friendly and fuel-efficient vehicles escalated.

On the other side, some researchers followed a more conservative approach to nonlinear observer design. Mostly inspired by the idea of extended Kalman filtering, the Luenberger observer was extended to nonlinear systems and its convergence properties were studied[29,30]. These extensions, under the influence of the classical observer design theory, were focused on analytical design techniques of the nonlinear observer. The main approach followed in this line of practice is to choose the observer gains such that the overall linearised error dynamics matrix consisting of the gain vector and the Jacobians of the state dynamics, and the output mapping has stable eigenvalues over a closed subset (0 of the state space. For convergence, the state trajectory is constrained to remain in this subset at all times[29,30]. This procedure of analytic ELO design has been applied successfully to realistic nonlinear systems[29-33]. It has also been utilised in designing nonlinear state feedback stabilisers for nonlinear systems [34,35]. In contrast to this analytical design approach, in this paper, we investigate the performance of the adaptive system approach to nonlinear observer design.

The organisation of this paper is as follows. In Section 2 we briefly describe Grossberg's additive model in the class of recurrent neural networks. Section 3 presents extensions to the Luenberger observer scheme for applicability to nonlinear systems and provides the link between the proposed adaptive observer scheme and the additive model. Section 4 describes the backpropagation algorithm for off-line training of the adaptive observer and Section 5 explains the necessary simplifications required to obtain an on-line training rule and offers the on-line training algorithm. Section 6 investigates the performance of the on-line trained adaptive observer on a variety of dynamical systems, including linear, nonlinear and chaotic systems. Also in Section 6 the application of the proposed adaptive observer scheme to a realistic partial engine model is considered. This is an important example for accurate estimation of the engine states and is imperative to efficient operation of the engine in terms of air-fuel ratio. Finally, we present our conclusions and proposed future lines of research related to the topic.

2. Additive Model for Recurrent Neural Networks

The most widely used dynamic neural network is the so-called additive model by Grossberg[36]. The state dynamics of the additive model is described by

$$\dot{x}(t) = -\tau\cdot(t) + \sigma\left(W_S\cdot x(t)\right) + W_I \cdot I(t) \tag{1}$$

Usually, the weights matrix multiplying the input vector $I(t)$ is chosen to be the identity matrix and the passive decay matrix τ is a diagonal positive definite matrix and the interactions between states is provided through W_S and the nonlinearity of the neurons, $\sigma(.)$, but these are not necessities. The biological motivation for the additive model provided by Sejnowski[37] has contributed to the increased popularity of this structure in numerous applications. The static MLP is just a special case of the additive model obtained by setting the time-derivative of the states to zero, thus imposing the staticity constraint on the states and restricting the weight matrix W_S to be strictly lower diagonal. In this case, the feedforward MLP is expressed as

$$\begin{aligned} x &= \tau^{-1}\sigma\left(W_S\cdot x\right)+\tau^{-1}\cdot W_I\cdot I \\ &= W_0\sigma\left(W_S\cdot x\right)+\overline{W}_I\ I \end{aligned} \tag{2}$$

where some of the states may be designated as the outputs of the MLP. A special case of interest is when the nonlinearity of the neurons in (1) is chosen to be a linear function. For the choice , (1) reduces to a linear dynamic neural network whose dynamics are of the form

$$\dot{x}(t) = \left(W_S - \tau\right)\cdot x(t) + W_I\cdot I(t) \tag{3}$$

In the following section we will point out how this relates to the classical Luenberger observer and thus provide an understanding of how the additive model in (1) connects to the extension that is proposed.

3. Nonlinear Extension of the Luenberger Observer

A well-known result from linear system theory is that, for a linear time-invariant (LTI) system with the dynamics

$$\begin{aligned} \dot{x}(t) &= A\cdot x(t) + B\cdot u(t) \\ y(t) &= C\cdot x(t) + D\cdot u(t) \end{aligned} \tag{4}$$

with an observable (A,C) pair, a stable linear Luenberger observer, which is given by

$$\dot{\tilde{x}}(t) = A\cdot\tilde{x}(t) + B\cdot u(t) + L\left(y(t) - \tilde{y}(t)\right) \tag{5}$$

can be designed by placing the poles of the observer at any desired location such that the error signals exhibit the desired dynamics[38]. The extension of the Luenberger observer to nonlinear systems is straightforward. Given a nonlinear dynamical system, possibly time-varying, whose equations are

$$\begin{aligned} \dot{x}(t) &= f\left(x(t),u(t),t\right) \\ y(t) &= h\left(x(t),u(t),t\right) \end{aligned} \tag{6}$$

the following observer scheme is utilised.

$$\begin{aligned} \dot{\tilde{x}}(t) &= f\left(\tilde{x}(t),u(t),t\right)+L\left(y(t) - \tilde{y}(t)\right) \\ \tilde{y}(t) &= h\left(\tilde{x}(t),u(t),t\right) \end{aligned} \tag{7}$$

Although there is a solid theory behind the linear Luenberger observer in (5) and there are rigorous analytical methods of selecting the observer gain vector L, such results are not available for the extended version in (7), yet. However, as we will demonstrate in the following sections, there is a way to overcome this difficulty by letting L adapt on-line while the system is running.

Regarding the link between the Luenberger observer structure and the additive model, we first point out the similarities between equations (3) and (5). In fact, if we re-express (5) in an alternative form, the connection becomes more visible. For this, we will substitute the explicit expression for $\tilde{y}(t)$ in the estimated state dynamics and then group the signals $u(t)$ and $y(t)$ into a vector.

$$\dot{\tilde{x}}(t) = \left(A - LC\right)\cdot\tilde{x}(t)+(B - LD)\cdot u(t)+L\cdot y(t) \qquad (8)$$
$$= (A - LC)\, x(t)+ \begin{bmatrix} B - LD & 0 \\ 0 & L \end{bmatrix} \cdot \begin{bmatrix} u(t) \\ y(t) \end{bmatrix}$$

Cross comparing (3) and (8), we conclude

$$\left(W_S - \tau\right) \equiv (A - LC) \qquad (9)$$
$$W_I \equiv \begin{bmatrix} B - LD & 0 \\ 0 & L \end{bmatrix}$$
$$I(t) \equiv \begin{bmatrix} u(t) \\ y(t) \end{bmatrix}$$

Thus, with proper adaptation, it is possible for the linear dynamical neural network to approximate a stable Luenberger observer for a linear system. In fact, we can generalise this sentence to the connection between the additive model in (1) and the extended Luenberger observer (ELO) in (7), since with sufficient number of neurons and proper choice of weight matrices the additive model can approximate any function with an arbitrarily small error. In fact, [19] demonstrates how approximately a neural network can learn the unknown system dynamics. In cases where a complete model of the dynamical system is not available, such approaches can be taken to obtain approximators of system models and substituted in the observer structure in proper places. From this point on, however, we will assume that either the full system dynamic equations are known or a neural network has been trained to sufficient accuracy as described for this purpose. This is because the main focus of this study is to determine the capabilities of the adaptive observer structure, not to investigate the function approximation capabilities of the mentioned additive model.

4. Backpropagation Algorithm for Extended Luenberger Observer

In this section we will consider the discrete-time equivalent of the ELO for reasons of analytical simplicity in computing and evaluating the gradient for off-line adaptation. In this context, the backpropogation refers to the backpropagation-in-time of the partial derivatives with respect to the observer gains. The system and the observer we are considering are given by the following equations.

$$x_{k+1} = f(x_k,u_k,k) \qquad (10)$$
$$y_k = h(x_k,u_k,k)$$

$$\bar{x}_{k+1} = f(\bar{x}_k,u_k,k)+L(y_k - \bar{y}_k) \qquad (11)$$
$$\bar{y}_k = h(\bar{x}_k,u_k,k)$$

Suppose we want to train for L such that the mean-square-error (MSE) along a given training trajectory $\{u_i,y_i\}_{i=0}^{N-1}$ is minimised. We would like to remark at this point that MSE is not the sole possibility as the performance criterion. In that case, the cost function and the gradient to optimise L for the given training trajectory using steepest descent algorithm are given below in (12). In fact, it has been previously shown that in many applications, information theoretic performance criteria outperform MSE and other second-order-statistics based criteria; mainly because second order statistics are not optimal anymore when the probability distributions involved are not Gaussian, and in order to achieve information-learning it requires more than just the second order statistics[39].

The gradient expression resembles the backpropagation of error in time that arises in the training of dynamical neural networks due to the dynamic observer structure. If the system in (10) is observable and LTI, only a single training trajectory is sufficient to obtain a globally asymptotically stable observer (the proof is omitted here). However, if this off-line mode training is assumed, then for nonlinear and time-varying systems, either the observer must be retrained at different points of the state-space or different observer gains trained for different locations in state-space or in time must be used switching from one observer to the other as the system moves in the state space.

$$J = \sum_{i=0}^{N-1}(y_i - \bar{y}_i)^T(y_i - \bar{y}_i) \qquad (12)$$

$$\frac{\partial J}{\partial L} = -2\sum_{i=0}^{N-1}(y_i - \bar{y}_i)^T h_x(\bar{x}_i,u_i,i)\frac{\partial \bar{x}_i}{\partial L}$$
$$\frac{\partial \bar{x}_i}{\partial L} = [f_x(\bar{x}_{i-1},u_{i-1},i\text{-}1) - L\, h_x(\bar{x}_{i-1},u_{i-1},i\text{-}1)]\,\frac{\partial \bar{x}_{i-1}}{\partial L}$$
$$+(y_{i-1}-\bar{y}_{i-1})\,[1]_{nxn}$$

$$L_{new} = L_{old} - \eta\frac{\partial J}{\partial L}$$

where $f_x(.)$ and $h_x(.)$ represent the Jacobians of the corresponding functions with respect to the state vector, and $[1]_{nxn}$ represents an all-ones square matrix of the size of the state vector.

In extensive simulations, this algorithm, which uses batch-training approach, was found ineffective in learning the dynamics and estimating the states of nonlinear systems, although it was very successful for LTI systems. In any case, an off-line training requirement may impose too much restriction to the applicability of an adaptive system to many tasks requiring real-time operation and adaptation. Therefore we would like to adopt an on-line training approach, thus avoid i) the requirement of off-line training, ii) the necessity of using multiple experts and switching. For these stated reasons, we modify the training approach from off-line to on-line, and employ Widrow's stochastic gradient approach.

5. Widrow's Stochastic Gradient Adaptation for Extended Luenberger Observer

When the instantaneous squared error is used as a stochastic approximation to MSE, the computed gradient of this stochastic cost function with respect to the weights, one gets Widrow's stochastic gradient for MSE[40]. Note that in computation of the stochastic gradient with respect to the weights of a recursive system Widrow suggests the designer approaches the problem with care and caution. Although the cost function depends only on the instantaneous value of the error, due to recursion the error still exhibits a backpropagation property and it is easy to oversee this. In the case of the ELO, at time step k, one may use the actual gradient expression computed with full consideration of the recursive structure of the topology or use an approximate version of the gradient, which is very accurate if the learning rate is chosen to be a small value. The former allows the use of larger learning rates, whereas the latter may go unstable for those same values of learning rates. The stochastic cost function and its approximate gradient (without consideration of the recursive nature of the system) are simply computed using (13) and the current value of the observer gains. For the full gradient expression that takes into account the recursive nature see Appendix A. Note that if the learning rate is chosen to be small, their behaviours are the same.

$$\hat{J} = (y_k - \bar{y}_k)^T (y_k - \bar{y}_k) \qquad (13)$$

$$\frac{\partial \hat{J}}{\partial L} = -2(y_k - \bar{y}_k)^T h_x(\bar{x}_k, u_k, k)\frac{\partial \bar{x}_k}{\partial L}$$

$$\frac{\partial \bar{x}_k}{\partial L} = [f_x(\bar{x}_{k-1}, u_{k-1}, k\text{-}1) \text{-} L\, h_x(\bar{x}_{k-1}, u_{k-1}, k\text{-}1)]\frac{\partial \bar{x}_{k-1}}{\partial L}$$

$$+(y_{k-1}\text{-}\bar{y}_{k-1})\,[1]_{nxn}$$

$$L_{k+1} = L_k - \eta\frac{\partial \hat{J}}{\partial L}$$

where η is the learning rate.

It is a well-known fact that Widrow's stochastic gradient algorithm makes the weights converge to the optimal MSE solution in the mean. Furthermore, the LMS algorithm is a well understood and proven algorithm that is useful in real-time adaptation problems[11] (yet, there are stochastic versions of the information theoretic adaptation criteria also, and in studies they are shown to exhibit all the advantageous properties of LMS and more[12]).

6. Illustrations With Basic Classes of Dynamical Systems

In this section, we will demonstrate the high performance of the adaptive ELO trained real-time with the stochastic gradient on a variety of dynamical systems, namely an LTI system, the Van der Pol oscillator, and the Lorenz attractor.

6.1 Linear Time-Invariant System

In the first case study, we consider a SISO LTI system excited by white Gaussian noise (WGN). The system matrices are given by

$$A = \begin{bmatrix} 0.9 & 0.1 \\ 0 & 0.8 \end{bmatrix}, b = \begin{bmatrix} 1 \\ -0.9 \end{bmatrix}, c = [1 \;\; 0] \qquad (14)$$

Two simulation results are presented, one with zero measurement noise and the other with WGN on the measurements.

Notice in *Figure 1*, where there is no measurement error, the (absolute value of the) state estimation errors decay exponentially, corresponding to an observer pole located inside the unit circle. In the noisy measurements case, we observe that the estimation errors cannot reduce to below a level determined by the signal-to-noise-ratio (SNR) of the measurements.

6.2 Van der Pol Oscillator

The second case study is a simple autonomous nonlinear system. The Van der Pol oscillator dynamics, discretised using the first order difference approximation are

$$x_{1,k+1} = x_{1,k} + T\,x_{2,k} \qquad (15)$$

$$x_{2,k+1} = x_{2,k} - 9T\,x_{1,k} + \mu\,T(1 - x_{1,k}^2)x_{2,k}$$

where we took the sampling time T=0.1 and the parameter μ = *0.5*. The system output is assumed to be the first state variable.

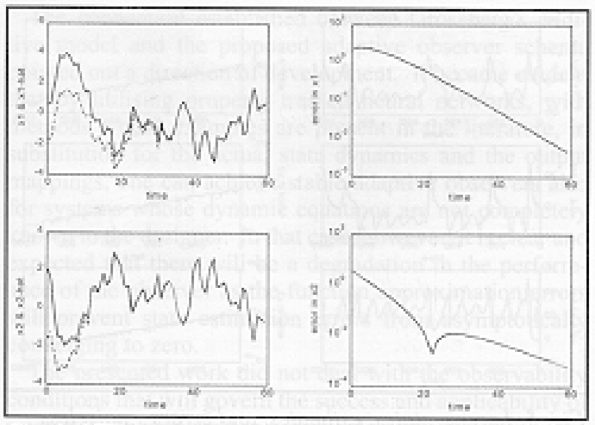

Fig. 1: LTI system states, estimates, and errors - noiseless.

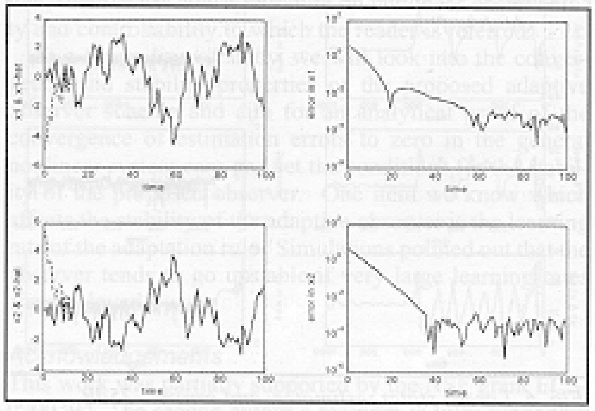

Fig. 2. LTI system states, estimates, and errors - 25dB.

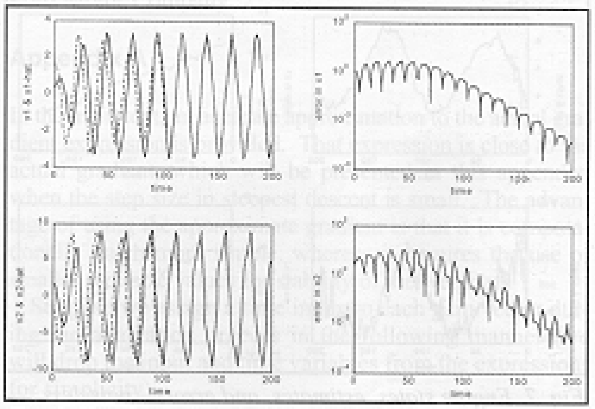

Fig. 3. Van der Pol states, estimates, and errors - noiseless.

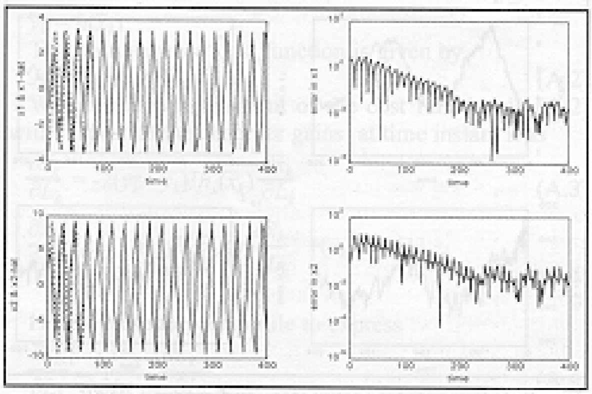

Fig. 4. Van der Pol states, estimates, and errors - 25dB.

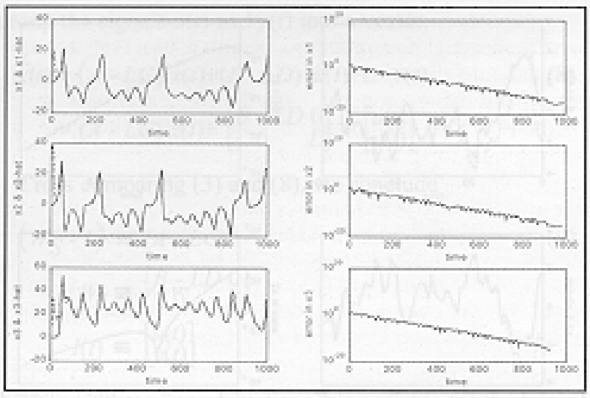

Fig. 5. Lorenz states, estimates, and errors - noiseless

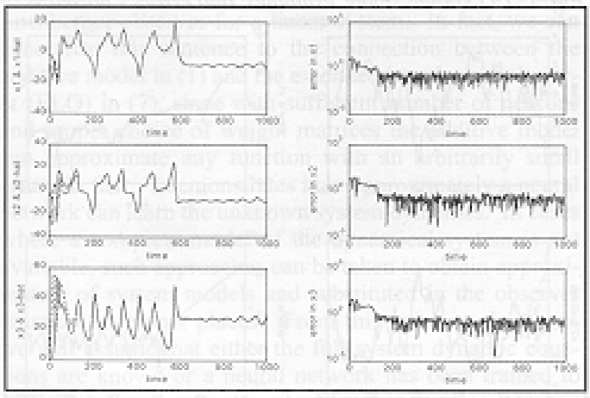

Fig. 6. Lorenz states, estimates, and errors - 25dB

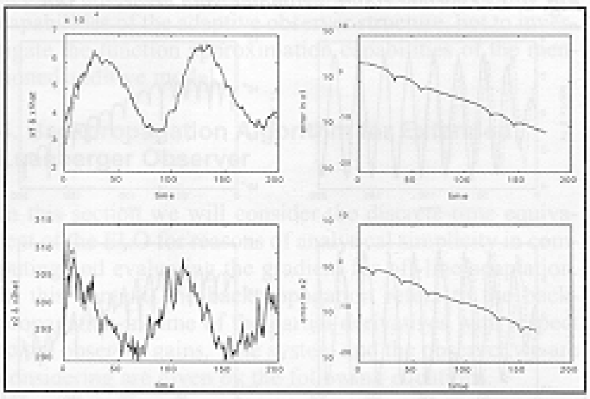

Fig. 7. Engine states, estimates, and errors - noiseless

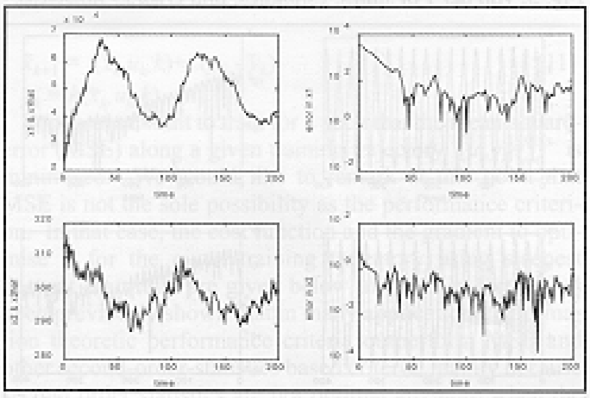

Fig. 8. Engine states, estimates, and errors - 25dB

Fig. 3 and *Fig. 4* show simulation results for the Van der Pol oscillator. Notice that the characteristic behaviour of the error remains and on the average, error decays exponentially when there is no noise and converges to a fixed value determined by the SNR when there is noise.

6.3 Lorenz Attractor

The Van der Pol oscillator states converge to a limit cycle and one suspects that it is this periodicity that helps the observer exhibit good performance in any way. In order to clear such doubts, we test the observer scheme on a chaotic system that has very high Lyapunov exponents, thus without any correctional terms, the slightest difference in initial conditions will lead to a very large divergence in the state trajectories. The Lorenz attractor dynamics, when discretised using the first order difference approximation for derivative, become

$$
\begin{aligned}
x_{1,k+1} &= (1-T\,\sigma)x_{1,k} + T\,\sigma\, x_{2,k} \\
x_{2,k+1} &= (1-T)\, x_{2,k} + T\, x_{1,k}\,(r - x_{3,k}) \qquad (16) \\
x_{3,k+1} &= (1-T\,b)\, x_{3,k} + T\, x_{1,k}\, x_{2,k}
\end{aligned}
$$

where the sampling time is taken as $T = 0.01$ and the parameters are chosen to be $\sigma = 10$, $r = 28$, $b = 8/3$. The system output is assumed to be the first state variable. *Figure 5* and *Figure 6* show simulation results for the Lorenz attractor. Once again, in this chaotic system case study, the adaptive observer performs successfully. Note that the adaptive observer utilises the linear correction term offered by the output error efficiently and by adapting its weights suitably, tracks the actual state vector accurately after an initial transient phase.

6.3 Realistic Engine Manifold Model

Mean value engine models are used to design AFR control systems in gasoline engines. These models are based on physical principles and some empirical correlations. They describe engine dynamics with limited bandwidth, equivalent to considering the mean behaviour of state variables over an engine cycle. A mean value model can be constructed either in time-domain or in crank-angle domain. It is also possible transform a time-domain model to crank-angle domain model or *vice-versa* by using the following relationship

$$dt = \frac{d\theta}{6N} \qquad (17)$$

where N is the engine speed in RPM and θ is crank angle in degrees. In the following a discrete mean value engine manifold model will be introduced in crank-angle domain. Similar models can be found in[27,43].

The intake manifold is placed between the throttle and the intake port. A two-state intake manifold model can be obtained from the conservation of energy and the mass in the manifold as

$$
\begin{aligned}
P_m(k+1) &= P_m(k) + \frac{T_s}{V_m}\left(\frac{(\gamma-1)}{6N}\; h_i(T_w - T_m(k)) \right. \\
&\qquad \left. +\gamma\, R(T_{amb} m_{ai}(k) - T_m(k) m_{ao}(k)\right) \qquad (18) \\
T_m(k+1 &= T_m(k) + \frac{T_s R T_m}{V_m P_m}\left(\frac{\gamma - 1}{6NR}\; h_i(T_w - T_m(k)) \right. \\
&\qquad \left. +(\gamma\, T_{amb} - T_m(k)) m_{ai}(k) + (1-\gamma) T_m m_{ao}(k)\right)
\end{aligned}
$$

where P_m is the manifold pressure, T_m is the manifold temperature, V_m is the manifold volume, γ is the specific heat ratio for air, h_i is the heat transfer coefficient, T_w is the manifold wall temperature, R is the specific gas constant for air, T_{amb} is the ambient air temperature, m_{at} is the throttle air flow rate, m_{ac} is the air flow rate to cylinder and T_s is the sampling time.

For a complete model of the engine for AFR control, fuel and sensor dynamics as well as the delays in measurements associated with sensors should be introduced to the model. However, we will not consider these extensions in this example. In the above two state model, the manifold pressure is assumed to be the measured output. *Figure 7* illustrates the states, estimations and errors for the noise-free measurements, where the throttle angle, the input, varies as a sinusoid with additive white noise in time. Note that the errors decay exponentially on the average. *Figure 8*, on the other hand, shows the results for the case where the measurements are corrupted with additive white noise. In that case, the estimation errors, as expected, decay to a value determined by the noise power in the measurements.

Although not reported here, the adaptive ELO is successfully applied to the complete engine model with eight states for the AFR control problem and it is observed that the observer-controller scheme manages to maintain the AFR at the optimal value, the stoichimoetric AFR.

7. Conclusions

The early works of Kalman and his contemporaries on the subject of observability and controllability of linear time-invariant systems form the basis of linear system theory. Their profound theory provides a deep understanding of linear system dynamics and offers analytical solutions to many problems, including state estimation, which is also the topic of the present study. However, in reality engineers have to face the difficulties that are associated with nonlinear systems, which are not tackled in the accomplished theory of linear systems. In order to overcome these difficulties, many solutions have been proposed. These solutions mainly concentrated around extensions through linearisation of nonlinear systems at operating points or application of adaptive/learning systems, with the main focus being on neural networks due to their inherent approximation and adaptability properties.

In this paper, we suggested the use of an adaptive extended Luenberger observer structure. Luenberger observer has a solid theory behind it for the case of linear time-invariant systems. The extension suggested allowed the use of the topology for nonlinear and time-varying systems without the necessity to solve for complicated analytical expressions. We had tested the performance of the adaptive extended Luenberger filter on a variety of systems ranging from chaotic to realistic models. It was found out that the proposed adaptive scheme was extremely successful in asymptotically estimating the states of the systems under examination when the system dynamic equations were completely known to the designer and was determined to be robust to noisy measurements. One advantage of this scheme over other adaptive methods is that the correction terms involved are linear and the structure of the observer is very simple. As a consequence the adaptation rules are extremely simple and require much less computation.

The connection established between Grossberg's additive model and the proposed adaptive observer scheme pointed out a direction of development. It became evident that by utilising properly trained neural networks, with methods whose examples are present in the literature, in substitution for the actual state dynamics and the output mappings, one can achieve stable adaptive observers also for systems whose dynamic equations are not completely known to the designer. In that case, however, it is clear and expected that there will be a degradation in the performance of the observer as the function approximation errors will prevent state estimation errors from asymptotically converging to zero.

The presented work did not deal with the observability conditions that will govern the success and applicability of asymptotic observers for a particular given system. For that, there exists a vast literature on nonlinear observability and controllability to which the reader is referred.

As a future line of study, we will look into the convergence and stability properties of the proposed adaptive observer scheme and aim for an analytical proof of the convergence of estimation errors to zero in the general nonlinear system case and set the conditions for the stability of the proposed observer. One item we know which affects the stability of the adaptive observer is the learning rate of the adaptation rule. Simulations pointed out that the observer tends to go unstable if very large learning rates are employed.

Acknowledgements

This work was partially supported by the NSF grant ECS-9900394. The second author's research is fully funded by Ford Motor Company.

Appendix A

In the main text, an accurate approximation to the actual gradient expression is provided. That expression is close to the actual gradient, which will be presented in this appendix, when the step size in steepest descent is small. The advantage of using the approximate gradient is that it is computationally much more simple, whereas it requires the use of smaller step size values for stability of the weights.

Suppose we assign a time index to each gain vector during the adaptation process in the following manner (we will drop the input and time variables from the expressions for simplicity).

$$\tilde{x}_{k+1} = f(\tilde{x}_k) + L_{k+1}(y_k - \tilde{y}_k) \qquad \text{(A.1)}$$

$$\tilde{y}_k = h(\tilde{x}_k)$$

The instantaneous cost function is given by

$$J_k = (y_k - \tilde{y}_k)^T (y_k - \tilde{y}_k) \qquad \text{(A.2)}$$

We compute the gradient of the cost function in (A.2) with respect to the observer gains at time instant k as

$$\frac{\partial J_k}{\partial L_k} = -2(y_k - \tilde{y}_k)^T h_x(\tilde{x}_k) \frac{\partial \tilde{x}_k}{\partial L_k} \qquad \text{(A.3)}$$

$$\frac{\partial \tilde{x}_k}{\partial L_k} \; [f_x(\tilde{x}_{k-1}) - L_k h_x(\tilde{x}_{k-1})] \frac{\partial \tilde{x}_{k-1}}{\partial L_k} + (y_k - \tilde{y}_k)[1]_{nxn}$$

Here we use the chain rule to express

$$\frac{\partial \tilde{x}_{k-1}}{\partial L_k} = \frac{\partial \tilde{x}_{k-1}}{\partial L_{k-1}} \cdot \frac{\partial L_{k-1}}{\partial L_k} \qquad \text{(A.4)}$$

Notice that when the step size is small, the second term on the right hand side of (A.4) will approximate to the identity matrix. This is the approximation that links this actual gradient expression to the approximate one given in (13). Now we can use the steepest descent update rule to detemine this second derivative. The matrix we seek is the inverse of

$$\frac{\partial L_k}{\partial L_{k-1}} = I - 2\eta\,(y_{k-1} - \tilde{y}_{k-1})h_x(\tilde{x}_{k-1})h_x(\tilde{x}_{k-2})^T \frac{\partial \tilde{x}_{k-2}}{\partial L_{k-2}} \frac{\partial L_{k-2}}{\partial L_{k-1}} \quad (A.5)$$

In summary, the actual gradient expression, which must take into account the recursive nature of the system, can be computed by iterating the equations presented in (A.3-5). They allow the use of larger learning rates, thus provide faster convergence rates at the cost of increased computational requirements.

References

[1] K. Hornik, M. Stinchoombe, H. White, "MLP's are universal approximators," *Neural Networks, vol. 2*, pp.359-366, 1989.

[2] D. Rumelhart, G. Hinton, R. Williams, "Learning internal representations by error backpropagation," *Nature, vol 323*, pp.533-536, 1986.

[3] F-L. Luo, U. Rolf, *Applied Neural Networks for Signal Processing*, Cambridge University Press, Cambridge, 1997.

[4] V.S. Cherkassky, F.M. Mulier, *Learning from Data: Concepts, Theory, and Methods*, Wiley, NY, 1998.

[5] J.A.K. Suykens, J.P.L. Vandewalle, B.L.R. Moor, *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*, Kluwer Academic Publishers, Netherlands, 1996.

[6] P.J. Werbos, "Neural Networks for Control and System Identification," *Proc. 28th CDC*, Tampa, FL,1989.

[7] R. Shoureshi, R. Chu, M. Tenario, "Neural Networks for system Identification," *Proc. Amer. Cont. Conf. vol. 1*, pp. 916-921, 1989.

[8] K.S. Narendra, K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. Neural Networks, vol. 1*, pp. 252-262, March 1990.

[9] M.M. Gupta, D.H. Rao (eds), *NeuroControl Systems: Theory and Applications*, IEEE Press, NY, 1994.

[10] W.T. Miller, R.S. Sutton, P.J. Werbos (eds), *Neural Networks for Control*, MIT Press, Cambridge, MA, 1995.

[11] R. Zhu, T. Chai, C. Shao, "Robust Nonlinear Adaptive Observer Design Using Dynamic Recurrent Neural Networks," *Proc. Amer. Cont. Conf.*, pp. 1096-1100, New Mexico, June 1997.

[12] S.S. Ge C.C. Hang, T. Zhang, "Adaptive Neural Networks Control of Nonlinear Systems by State and Output Feedback," *IEEE Trans. Sys., Man, and Cyb.-Part B: yb., vol. 29, no. 6*, Dec. 1999.

[13] J.A.Ruiz Vargas, E.M. Hemerly, "Neural adaptive observer for general nonlinear systems," *Proc. Amer. Cont. Conf.*, pp. 708-712, Chicago, IL, June 2000.

[14] F. Sun, Z. Sun, P-Y. Woo, "Neural Network-Based Adaptive Controller Design of Robotic Manipulators with an Observer," *IEEE Trans. Neural Networks, vol. 12*, no. 1, pp. 54-67,Jan. 2001.

[15] G.V. Puskorius, L.A. Feldkamp, "Neurocontrol of Nonlinear Dynamical Systems with Kalman Filter Trained Recurrent Networks," *IEEE Trans. Neural Networks, vol. 5, no. 2*, pp. 279-297, Mar. 1994.

[16] J. Wang, G. Wu, "Real-Time Synthesis of Linear State Observers Using a Multilayer Recurrent Neural Network," *Proc. IEEE Int. Conf. Industrial Tech.*, pp. 278-282, 1994.

[17] M.S. Ahmed, S.H. Riyaz, "Design of Dynamic Neural Observers," *IEE Proc. Cont. Theory Appl., vol. 147*, n0. 3, pp.257-266, May 2000.

[18] T. Fretheim, R. Shoureshi, T. Vincent, "A General Approach to Non-Linear Output Observer Design Using Neural Network Models," *Proc. of Amer. Cont. Conf.*, pp. 924-928, June 2000.

[19] N. Hovakimyan, H. Lee, A. Calise, "On Approximate NN Realization of an Unknown Dynamic System from its Input-Output History," *Proc. Amer. Cont. Conf.* pp.919-923. Chicago, IL, June 2000.

[20] Y. H. Kim, F.L. Lewis, "Neural Network Output Feedback Control of Robot Manipulators," *IEEE Trans. on Robot. And Autom., vol. 15, no. 2*, pp. 301-309., Apr. 1999.

[21] F-J. Lin, R-J. Wai, C-H. Lin, D-C. Liu, "Decoupled Stator-Flux-Oriented Induction Motor Drive with Fuzzy Neural Network Uncertainty Observer," *IEEE.Trans. Industrial Electronics, vol. 47, no. 2*, pp. 356-367, Apr. 2000.

[22] P. Marino, M. Milano, F. Vasca, "Robust Neural Network Observer for Induction Motor Control," *Power Electronics Specialists Conf.*, pp 699-705, 1997.

[23] J. Theocharis, V. Petridis, "Neural Network Observer for Induction Motor Control," *IEEE Cont. Sys. Mag. vol. 14, no. 2*, pp. 26-37, Apr. 1994.

[24] F-J. Lin, C-H. Lin, C-M. Hong, "Robust Control of Linear Synchronous Motor Servodrive Using Disturbance Observer and Recurrent Neural Network Compensator," *IEE Proc. Electr. Power, vol. 147, no. 4*, pp. 263-272, July 2000.

[25] S. Pillutla, A. Keyhani, I. Kamwa, "Neural Network Observers for On-Line Tracking of Synchronous Generator Parameters," *IEEE Trans. Energy Conv. Vol. 14, no. 1*, pp. 23-30, March 1999.

[26] A.M. Frith, C.R. Gent, A.J. Beaumont, "Adaptive Control of Gasoline Engine Air-Fuel Ratio Using Artificial Neural Networks," *Proc. IEE Conf. Artificial Neural Networks*, pp. 274-278, June 1995.

[27] J.D. Powell, N.P.Fekete, C-F. Chang, "Observer-Based Air-Fuel Ratio Control," *IEEE Cont. Sys. Mag., vol. 18, no. 5*, pp. 72-83, Oct. 1998.

[28] C. Manzie, M. Palaniswami, H. Watson, "Model Predictive Control of a Fuel Injection System with Radial Basis Function Network Observer," *Proc. Int. Joint Conf. Neural Networks*, pp. 359-364, 2000.

[29] C. Elmas, H. Zelaya-de la Parra, "Application of a full-order Luenberger observer for a position sensorless operation of a switched reluctance motor drive," *IEE Proc. Control Theory Appl., vol 143, no 5*, Sept. 1996.

[30] T. Du, T., P. Vas, F. Stronach, "Design and application of extended observers for joint state and parameter estimation in high performance AC drives," *IEE Proc. Electr. Power. Appl., vol 142, no 2*, Mar. 1995.

[31] T. Orlowska-Kowalska, "Application of extendd Luenberger observers for flux and rotor time-constant estimation in induction motor drives," *IEE Proceedings, vol. 136, no. 6*, Nov. 1989.

[32] J. Song, K-B. Lee, J-H. Song, I. Choy, K-B. Kim, "Sensorless vector control of induction motor using a novel reduced order extended Luenberger observer," *Conf. Record of the 2000 Industry Appl. Conf., vol. 3*, pp. 1828-1843, 2000.

[33] T. Du, M.A. Brdys, "Implementation of extended Luenberger observers for joint state and parameter estimation of PWM induction motor drive," *5th European Conf. on Power Elctr. Appl., vol. 4*, pp. 439-444, 1993.

[34] J. Rodrigues-Millan, Y. Alzuru, J. Cardillo, J. Yepez, "NL Feedback 2.0: a symbolic computation tool for the design of extended controllers and observers for nonlinear control systems with stabilizable and detectable linearization," *Proc. Amer. Cont. Conf., vol. 3*, pp.1390-1394, 1997.

[35] C. Delepaut, G. Bastin, M. Gevers, "Stabilization of nonlinear systems by means of state estimate feedback," *Proc. 28th IEEE Conf. Decision and Cont., vol. 2*, pp. 1042-1046, 1989.

[36] J.C. Principe, N. Euliano, C. Lefebvre, *Neural and Adaptive Systems*, John Wiley and Sons Inc, NY, 1999.

[37] T.J. Sejnowski, "Skeleton Filters in the Brain," in *Parallel Model of Associative Memory*, pg. 189-212, Hinton and Anderson (eds), LEA Inc., 1981.

[38] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[39] D. Erdogmus, J.C. Principe, "Entropy Minimization Algorithm for Multilayer Perceptrons," to appear in *Proc. Int. Joint Conf. on Neural Networks*, July 2001.

[40] B. Widrow, S.D.Stearns, *Adaptive Signal Processing*, Prentice Hall, NJ, 1985.

[41] S. Haykin, *Adaptive Filter Theory*, 3rd ed., Prentice Hall, Inc., NJ, 1996.

[42] D. Erdogmus, J.C. Principe, "An On-Line Adaptation Algorithm For Adaptive System Training with Minimum Error Entropy: Stochastic Information Gradient," submitted to *Independent Component Analysis 2001*.

[43] A. Chevallier, C.W. Vigild, E. Hendricks, "Predicting the Port Air Mass Flow of SI Engines in Air-Fuel Ratio Control Applications", 2000-01-0260, *SAE 2000 World Congress*.