Computers Math. Applic. Vol. 22, No. 1, pp. 67–75, 1991
Printed in Great Britain.

0097-4943/91 $3.00 + 0.00

AN ADAPTIVE LEARNING APPROACH TO THE IDENTIFICATION OF STRUCTURAL AND MECHANICAL SYSTEMS

ROBERT E. KALABA
Departments of Biomedical Engineering, Electrical Engineering and Economics
Olin Hall, University of Southern California, Los Angeles, CA 90089-1453
FIRDAUS E. UDWADIA
Departments of Mechanical Engineering, Civil Engineering and Decision Systems
Olin Hall, University of Southern California, Los Angeles, CA 90089-1453

(Received October 1990)

Abstract—The identification of parameters in models of structural and mechanical systems is an important problem. The usual approaches are successive approximation schemes which require good initial guesses for rapid convergence. This paper shows how such initial approximations may be obtained. Notions from the field of artificial neural networks are used. In fact, new adaptive schemes for learning are presented and used in parameter estimation for both linear and nonlinear systems.

1. INTRODUCTION

Parameter identification of structural and mechanical systems has become a very important field in the area of applied mechanics [1,2]. This is motivated by the need to accurately predict the response of such systems and/or to be able to control them so that they can perform their functions adequately [2]. Areas such as the adaptive control of robot manipulators, and structural vibration control of mechanical, aerospace and civil engineering structures [3–5], require that the parameters that model the physical system be identified appropriately. Most of the time, parameter identification, even for linear mechanical systems, results in a nonlinear optimization problem requiring an iterative approach to its solution [6,7]. Several on-line and off-line methods have been developed in the past, most of them based on different variants of the Gauss-Newton method. Probabilistic formulations such as maximum likelihood and the extended Kalman filter have been developed as also continuation and homotopy methods [1,7–9]. Yet, the basic problem, when facing the identification of a large number of parameters from input-output data, is caused by the fact that the objective function surface may have multiple minima and therefore convergence to the correct parameters is iteratively possible only when one starts from a close enough initial guess of the parameters to be identified [2,10]. Furthermore, for large spatially extended complex systems the nature of the inverse problem has in it the inherent possibility of yielding nonunique parameter estimates when using response data obtained from a few locations in the system; this, even in the absence of measurement noise [11].

An alternative method of performing such system identification is to use an associative memory approach [12,13]. Here, rather than iteratively solving the inverse problem for a given input-response pair as is commonly done in on-line identification, the forward problem is repeatedly solved for various input-response pairs, and a memory matrix is adduced which optimally associates the inputs with the outputs. The identification method relies on providing sufficient 'training' (i.e., exposure to different input-response pairs) for an adequate knowledge base to be acquired. Thus when the identification scheme is later presented with a given input (output), it can then estimate the output (input) that corresponds to it. Different learning schemes that develop the memory matrix that maps the input to the output have been developed. The literature on neural networks, linear associated memories and bidirectional associative memories provides procedures for doing this [1,6,12–17].

Yet, the underlying approach provided in this paper differs significantly from those so far utilized in the neural network literature. When viewing a dynamic system via a neural net model, the system's model is primarily encapsulated in the "connections weights" (which are functions of the parameters to be identified). These are usually sought by training the system through an appropriate set of input-output training pairs [17]. For mechanical systems, the unknown physical parameters, like the stiffness and inertia of a system, are related to the weights; various dynamic loads are provided to the system, and its time history of response to these loads is used for training the network. Most of the approaches used to date in structural applications perform history matching and iteratively update the weights of the neural model, a procedure often tantamount to performing a gradient search. For example, Rehak *et al.* [18] have applied the simple back propagation method to identify the parameters of a simple mechanical linear dynamic system using a given set of force inputs and response pairs. The weights were functions of the stiffness, the damping and the inertial properties of the system. They were unable to obtain any meaningful results despite the large number of learning steps that they used.

In the approach used here the learning 'input' to the system constitutes directly the system parameters. This has two important advantages: (1) we always solve forward problems, which are usually much easier to handle than their inverse counterparts; (2) the behavior of these forward solutions is better understood than those of inverse problems; and (3) we can parameterize the dynamic loading, thereby obtaining estimates of *both* the input to the system as well as its parameters (see Example 1 in Section 3 below). It is this conceptual difference involving what constitutes an 'input' to the system for purposes of training it, that we believe is responsible for the unusually good performance of the technique to be described herein in comparison with some of the results of previous investigators that have used procedures that may fall under rubric of neural nets.

We consider here systems whose models we assume are known in structure. The values of the parameters describing the models need identification. Furthermore we assume that training is constituted by providing to the identification scheme, sets of parameter values and the respective time histories of responses. The dynamic inputs are taken to be known or described by parameters which also need to be identified. Thus for each vector of parameters p_i in a training set $\{p_1\, p_2\, p_3 \ldots p_n\}$ we determine, on the basis of the model we have, the corresponding response vector r_i. The set of responses $\{r_1\, r_2\, r_3 \ldots r_n\}$ along with the set $\{p_i\}$ are then used to generate a memory matrix M. Then, for a given observed response vector r^*, an estimate of the corresponding parameter vector p^* is obtained by setting $p^* = Mr^*$. While not much may be expected from such a linear approximation to a highly nonlinear relation between p and r, our numerical results show that the idea of a linear associative memory provides dramatically accurate parameter identification through the use of a very nominal amount of training.

In [14], surprisingly accurate estimates of the parameters were obtained by using a linear associative memory matrix, $M = PR^+$, where the matrix $P = [p_1\, p_2\, p_3 \ldots p_n]$ and the matrix R^+ is the Moore-Penrose [19] generalized inverse of the matrix $R = [r_1\, r_2\, r_3 \ldots r_n]$. However when noisy observations were used to estimate the parameters using memory matrices obtained from noise-free data, instabilities were found to occur. Furthermore the presence of elements having large orders of magnitude relative to the components of the training vectors were found to make the parameter estimates extremely sensitive to noise; under these conditions computational round-off errors were found to have a considerable effect on the results, even in the absence of noise. The ill-conditioning of the memory matrix was reduced in [15] by the use an additional criterion, namely, to make the elements of M small. This multi-criterion approach was found to perform well on the identification of the coefficients of a nonlinear ordinary differential equation that modelled the growth of an economy at the macro level. A large set of training pairs was used; the main purpose of the study was to generate improved initial estimates using associative memory methods for parameter identification, so that these initial estimates could then be used in standard identification procedures like steepest descent, conjugate gradient or MLE formulations.

In this paper we use the multicriterion approach developed in [15] and extend it to adaptively train the identification scheme. While we consider our findings preliminary, we show that such adaptive training can be used very effectively, not just in obtaining initial parameter estimates but in parameter identification of mechanical systems that can be modelled by differential equations.

By starting out with just a few training pairs of inputs and responses, and adaptively generating more pairs, one can obtain very good estimates of the parameter vectors to be identified. Numerical results are shown for two systems, one linear and the other nonlinear, which commonly arise in civil and mechanical engineering. Section 2 of the paper provides a short formulation and derivation of the memory matrix and the adaptive scheme used herein, and Section 3 applies the technique to a problem of identifying the parameters of a system modelled by a single degree of freedom oscillator, as well as a system described by a nonlinear differential equation. In Section 4 we present our conclusions and several issues that the numerical results raise.

2. DETERMINATION OF THE MEMORY MATRIX AND THE ADAPTIVE LEARNING SCHEME

a) Estimation of the Memory Matrix

Consider a set of n training inputs and responses such that each input p_i consists of a vector belonging to $\mathbf{R}^u$, and each response r_i is a vector belonging to $\mathbf{R}^v$. We seek to linearly associate the responses to the inputs using a memory matrix M such that the error of association is as small as possible. To prevent M from being ill-conditioned we need also to limit the size of its elements. We thus construct a multicriterion cost function [15]

$$J = \alpha \, ||M\,R - P||^2 + (1-\alpha)\,||M||^2, \qquad 0 \leq \alpha \leq 1, \tag{1}$$

where, the parameter α indicates the relative emphasis on the the cost associated with mapping the input to the output, and the cost of M being ill-conditioned. The matrices R and P are $u \times n$ and $v \times n$, respectively. The euclidean matrix norm of M is defined as

$$||M|| = \sqrt{\mathrm{Tr}\{M^T M\}}. \tag{2}$$

Minimizing J with respect to the elements of M, we obtain

$$\hat{M} = P\,\left[\alpha\,R^T\,(\alpha\,RR^T + (1-\alpha)\,I)^{-1}\right] = PR^{\sim}. \tag{3}$$

When $\alpha = 1$, the matrix $R^{\sim}$ becomes the generalized inverse R^{+}, and the sensitivity to noise and computational round-off increases. As the value of α reduces, the ill-conditioning of $\hat{M}$ reduces but so does the error in associativity between the input and the response (see [15] for a numerical substantiation of these statements).

b) Estimation of the Parameter Vector

Having obtained $\hat{M}$ using the set of n training pairs, the estimate of p^* corresponding to an observation r^* then becomes

$$\hat{p}_n^* = \hat{M}_n\,r^*, \tag{4}$$

where the subscript n indicates that this estimate is predicated upon the use of n prior training pairs $\{p_1\,p_2\,p_3 \ldots p_n\}$ and $\{r_1\,r_2\,r_3 \ldots r_n\}$.

Now consider the mapping that maps the parameter vector p to the response, r, namely $f : p \rightarrow r$. Assuming that a local inverse is available, we then have $p = f^{-1}(r)$, which we assume can be approximately expressed by a memory matrix by the relation $p = Mr$. But we only have an *estimate* of the memory matrix, M_n, so that

$$\hat{M}_n = M_n + \delta\,M_n, \tag{5}$$

and hence the estimated parameter vector in (4) is expressible as

$$\hat{p}_n^* = (M_n + \delta\,M_n)\,r^*. \tag{6}$$

Denoting the true parameter vector corresponding to the response r^* by $\hat{p}_n^*$ we obtain

$$\hat{p}_n^* = M_n r^*, \tag{7}$$

and

$$\bar{p}_n^* = \hat{p}_n^* + \delta M_n r^* = (M_n + \delta M_n) r^*. \tag{8}$$

Now if we were to use this estimate $\bar{p}_n^*$ to obtain $\bar{r}_n^*$ (i.e., solve of the forward problem), we would have, locally,

$$\bar{p}_n^* = M_n \bar{r}_n^*. \tag{9}$$

Using (9) in (8) we then get

$$\delta M_n r^* = M_n (\bar{r}_n^* - r^*) := M_n \{\delta r_n^*\}. \tag{10}$$

From this we finally obtain an improved estimate

$$\hat{p}_n^* = \bar{p}_n^* - M_n \{\delta r_n^*\}. \tag{11}$$

However the memory matrix M_n is not available. Noting (5), the updated estimate of the parameter vector then becomes

$$\hat{p}_n^* = \bar{p}_n^* - (\hat{M}_n - \delta M_n)\{\delta r_n^*\} \cong \bar{p}_n^* - \hat{M}_n \{\delta r_n^*\}, \tag{12}$$

where we have assumed that the quantity $\delta M_n \cdot \delta r_n^*$ is small.

c) Adaptive Creation of Training Vectors

Having thus obtained an updated estimate $\hat{p}_n^*$ using the available training set, we now generate a training pair using a parameter vector in the vicinity of this estimate, $\hat{p}_n^*$. Thus we adaptively generate the next training input vectors as

$$p_{i+1}^{(j)} = \hat{p}_i^{(j)^*} (1 + \beta_i \xi_i^{(j)}), \qquad i = n, n+1, \ldots, n+q; \quad j = 1, 2, \ldots, u, \tag{13}$$

where $p_{i+1}^{(j)}$ is the jth component of vector p_{i+1}, $\xi_i^{(j)}$ is a random number, uniformly distributed between -0.5 and $+0.5$, and β_i is a scaling factor. For each adaptively generated training vector p_{i+1}, $i \geq n$, a corresponding response vector r_{i+1}, $i \geq n$, is obtained. This input-response pair is added to the set of available training pairs and the memory matrix is updated, yielding $\hat{M}_{i+1}$, $i \geq n$, along with a new estimate $\bar{p}_{i+1}^*$. This estimate is then updated as described above to give $\hat{p}_{i+1}^*$. Since each random training vector samples a slightly different region of the parameter space, the final parameter vector may then be taken to be the average over the $(q+1)$ estimates obtained.

Algorithmically then we have:

Step 1: Obtain an updated estimate $\hat{p}_n^*$ using the training set S_n (which contains n input-response pairs) by:
- (a) using relation (4) to obtain $\bar{p}_n^*$,
- (b) solving the forward problem to determine $\bar{r}_n^*$,
- (c) thence finding δr_n^* and, finally,
- (d) using relation (12) to get $\hat{p}_n^*$.

Step 2: For $i = n$ to $n + q$
do
1. Generate a close-by input p_{i+1} using equation (13);
2. Generate the corresponding response r_{i+1};
3. Add this newly generated input-response pair to the set S_i producing the set S_{i+1} which has $i + 1$ pairs on inputs and outputs;
4. Use the set S_{i+1} and relation (3) to generate $\hat{M}_{i+1}$;
5. Obtain an estimate $\hat{p}_{i+1}^*$ using $\hat{M}_{i+1}$ as in Step 1.

end.

Step 3: Take the average of the vectors $\hat{p}_i^*$, $i = n, n+1, \ldots, n+q$.

3. NUMERICAL EXAMPLES

Consider a structure described as single degree-of-freedom damped oscillator governed by the differential equation

$$\ddot{x}(t) + 2\omega_n \xi \dot{x}(t) + \omega_n^2 x(t) = 0, \qquad x(t=0) = x_0, \quad \dot{x}(t=0) = \dot{x}_0.$$

We shall assume that all quantities are in consistent units. We consider the identification of the parameters ω_n, ξ, and $\dot{x}_0$, starting with a set of 5 training inputs. The true parameter vector is $[8\ 0.09\ 20]^T$. The 5 training vectors use parameters which roughly (heuristically) cover the region of the true parameter vector. They are:

$$[6, 0.02, 10]^T, \quad [20, 0.06, 24]^T, \quad [16, 0.03, 18]^T, \quad [10, 0.01, 25]^T, \quad \text{and } [7, 0.10, 15]^T.$$

The initial displacement x_0 is taken to be zero. We use 20 response data points at various times, t, for the determination of the memory matrix $\hat{M}_5$. Of the twenty response data used, 10 data points were gathered starting at zero time with a spacing of 0.02 units and an additional 10 were acquired starting at time 3.3 units with a time spacing of 0.3 units. The parameter α is chosen to be 0.9 and the parameters $\beta_i := \beta = 0.5$, for all i.

We obtain the estimate $\hat{p}_5^*$ of the parameter vector $[\omega_n\ \xi\ \dot{x}_0]$ and then successively generate random vectors in the neighborhood of this estimate. Twenty such additional nearby random training vectors are used. The resulting initial estimate $\hat{p}_5^*$, and the final estimate after adding the additional training vectors are shown in Table 1a. We observe that through the use of the additional training vectors a considerable improvement in the parameter estimates is obtained and the method with just 20 data points has been able to identify the system rather accurately. Note that parameters related to *both* the input and the system are being simultaneously identified.

Table 1a. Noise to signal ratio = 0%. 10 data points $\in$ [0,0.2] time units; 10 data points $\in$ [3.3,6] time units; $\alpha = 0.9$; $\beta = 0.50$.

True parameter values	Estimate using 5 training vectors	% Error	Estimate using 25 training vectors	% Error
$\omega_n = 8.0$	10.56	−32.05%	8.29	−3.64%
$\xi = 0.09$	0.106	−18.06%	0.088	2.32%
$\dot{x}_0 = 20.0$	19.675	1.62%	19.94	0.3%

Table 1b. Noise to signal ratio = 0%. 20 data points $\in$ [0,0.4] time units; $\alpha = 0.9$; $\beta = 0.50$.

True parameter values	Estimate using 5 training vectors	% Error	Estimate using 25 training vectors	% Error
$\omega_n = 8.0$	8.85	−10.7%	7.81	2.33%
$\xi = 0.09$	0.053	−40.75%	0.034	61.7%
$\dot{x}_0 = 20.0$	19.60	1.62%	19.94	0.3%

The reason why the data was gathered over two disjoint time intervals is that the initial response of the system is sensitive to the first and third parameters to be estimated while longer time responses provide greater detail on the damping parameter. Table 1b shows that the collection of 20 data points spaced at 0.02 time units apart beginning with zero time, while yielding reasonable results for the first and third parameters does not provide a good estimate for the damping parameter, as expected. The choice of the times at which the data is sampled for use in the identification scheme is thus crucial to its efficacy.

The effect of noise is indicated in Tables 2a and 2b, where to each response is added a zero mean random i.i.d. variable. Noise-to-signal ratios of 6% and 12% are used. We observe that

Table 2a. Noise to signal ratio = 6%. 10 data points $\in$ [0,0.2] time units; 10 data points $\in$ [3.3,6] time units; $\alpha = 0.9$; $\beta = 0.50$.

True parameter values	Estimate using 5 training vectors	% Error	Estimate using 25 training vectors	% Error
$\omega_n = 8.0$	9.84	−23.67%	8.06	−0.867%
$\xi = 0.09$	0.102	−13.54%	0.123	37.30%
$\dot{x}_0 = 20.0$	18.75	6.24%	20.38	−1.92%

Table 2b. Noise to signal ratio = 12%. 10 data points $\in$ [0,0.2] time units; 10 data points $\in$ [3.3,6] time units; $\alpha = 0.9$; $\beta = 0.50$.

True parameter values	Estimate using 5 training vectors	% Error	Estimate using 25 training vectors	% Error
$\omega_n = 8.0$	9.84	−23.67%	8.9	−11.29%
$\xi = 0.09$	0.102	−13.54%	0.13	−50.33%
$\dot{x}_0 = 20.0$	18.75	6.24%	20.13	−0.672%

the numerical values are not significantly affected, though the percentage errors have increased when compared with the no-noise situation of Table 1a.

Table 3 shows the results of identifying all four parameters, $[\omega_n\ \xi\ \dot{x}_0\ x_0]^T$. The true parameter vector is taken to be $[8.25\ 0.07\ 10.0\ 7.5]^T$. Starting with the four training parameter vectors

$$[7, 0.06, 12, 6]^T, \quad [9, 0.08, 8, 8]^T, \quad [7, 0.06, 10, 7]^T, \quad [9, 0.05, 11, 9]^T,$$

the estimate $\hat{p}_4^*$ is generated and further estimates are obtained using an additional twenty randomly generated training vectors. The values of $\alpha = 0.9$ and the parameters $\beta_i := \beta = 0.2$ are for all i. Two sets of data are used for the identification. The first set consists of the response at twenty time points spaced at 0.02 time units. The second set consists of thirty points, with again a spacing of 0.02 time units. Improvement in the identified parameters using an increased amount of response data is noted, despite the fact that the initial estimate, $\hat{p}_4^*$, is somewhat better in the former case.

Table 3. Noise to signal ratio = 0%. $\alpha = 0.9$; $\beta = 0.50$.

	20 data points $\in$ [0,0.2] time units		30 data points $\in$ [0,0.3] time units	
True parameter values	Estimate using 25 training vectors	% Error	Estimate using 25 training vectors	% Error
$\omega_n = 8.25$	8.13	1.45%	8.24	0.107%
$\xi = 0.07$	0.059	15.18%	0.0617	11.9%
$x_0 = 10.0$	9.62	3.7%	10.27	−2.78%
$\dot{x}_0 = 7.5$	7.51	−0.12%	7.47	0.28%

Table 4 shows some results on the identification of four parameters of the nonlinear differential equation

$$\ddot{x}(t) + p_1\,\dot{x}(t) + p_2\,x(t) + p_3\,x^3(t) + p_4\,\dot{x}(t)\,|\dot{x}(t)| = a\,\sin(b\,t), \qquad x(0) = \dot{x}(0) = 0,$$

where the parameters a and b are taken to be 10 and 3.3 in appropriate units. The parameters p_i, $i = 1, 2, 3, 4$, on the left hand side of the equation are estimated starting with a set of 4 training vectors. The true parameter values are $[0.06\ 10\ 2\ 0.07]^T$. The training vectors used are:

$$[0.01, 10, 3, 0.05]^T, \quad [0.08, 12, 1, 0.09]^T, \quad [0.05, 14, 5, 0.02]^T, \quad [0.03, 8, 4.0, 0.06]^T.$$

Five additional training vectors are generated as described in the previous section using in each case a β value of 0.5. Twenty response data points are gathered at a spacing of 0.10 time units. This response data was obtained by numerically integrating the differential equation using 4th order Runge-Kutta with an integration tolerance of 0.005 units. As seen from Table 4 the results are fairly good considering that only a few training vectors and a relatively small amount of response data were used. There is little improvement, in this case, in the parameter estimates when the number of data points is doubled from 20 to 40 over the same time interval. As seen from Table 4, for smaller values of α, the convergence is somewhat slower, and the estimates worsen for comparable numbers of training pairs.

Table 4. Noise to signal ratio = 0%.

	20 data points; $\alpha = 0.97$		20 data points; $\alpha = 0.80$	
True parameter values	Estimate using 9 training vectors	% Error	Estimate using 9 training vectors	% Error
$p_1 = 0.06$	0.049	19.17%	0.0460	23.26%
$p_2 = 10.0$	10.662	−6.69%	9.9072	0.9278%
$p_3 = 10.0$	2.132	−6.60%	2.20	−10.058%
$p_4 = 0.07$	0.073	−3.91%	0.071	−1.54%

Table 5a shows the seven initial trial vectors used in the identification of all the parameters of the nonlinear system, including the parameters a and b which describe the forcing function. Table 5b shows both the initial estimates obtained using these seven vectors, and the final results obtained using a total of 17 trial vectors, 10 of them randomly selected according to the algorithm

Table 5a.

Trial	Parameters					
vector	p_1	p_2	p_3	p_4	a	b
1	0.01	10.0	3.0	0.05	13.0	4.0
2	0.08	12.0	1.0	0.09	8.0	3.5
3	0.05	14.0	5.0	0.02	10.0	4.0
4	0.03	8.0	4.0	0.06	9.0	3.0
5	0.04	8.0	3.0	0.05	11.25	3.5
6	0.07	9.0	6.0	0.02	9.25	2.5
7	0.06	12.0	2.50	0.08	11.0	3.0

Table 5b. Forty data points belonging to [0,4]; $\alpha = 0.9$; $\beta = 0.3$.

Exact parameter values	Estimate using 7 training vectors	% Error	Estimate using 17 training vectors	% Error
$p_1 = 0.06$	0.0795	32.45	0.0557	7.19
$p_2 = 10.0$	14.118	−41.18	10.03	−0.31
$p_3 = 10.0$	4.04	−102.0	1.98	1.03
$p_4 = 10.0$	0.102	−46.41	0.0829	−18.45
$a = 10.0$	10.66	−6.5	9.8756	1.24
$b = 3.3$	4.17	−26.43	3.44	−4.28

proposed herein. The equation is integrated using the fourth order Runge-Kutta method with an integration (local) error tolerance of 0.01 units. Forty equi-spaced response data points in the interval $t \in [0, 4]$ are used. The values of α and β are 0.9 and 0.3, respectively. The results indicate that the method may be able to simultaneously yield reasonably good estimates of both the system parameters as well as the parameters that describe the forcing function.

It should be noted that the results that have been obtained in this section rely on the generation of adaptive training vectors; as this is done with the help of a random number generator, for each random sequence of computer generated numbers the resulting parameter estimates will be somewhat different. Estimation of the parameters using several different random 'seed' numbers was carried out and the results shown in Tables 1 through 5 appear to be typical in terms of the percentages of error found.

Convergence was found to be controlled primarily by the choice of the initial training set, and by the parameters α and β. For large values of β the training vectors sometimes had large excursions leading at times to a divergence of the estimate; on the other hand small excursions, caused by small values of β, led to slow convergence.

The times at which response data was gathered also had a strong influence on the quality of the parameter estimates obtained (see Table 1). Besides this, it was found that large response data vectors often led to numerical problems because the matrices were ill-conditioned.

The updating procedure described in Section 2(b) was found to be very important in bringing about a quick convergence of the procedure. In fact without such updating the adaptive scheme was found to diverge in most of the examples illustrated herein. This is especially true when the number of initial training vectors used is small, making the initial estimates of M rather inaccurate.

4. CONCLUSIONS

In this paper we have explored a simple method for parameter identification which relies on starting with a set of training inputs and adaptively generating additional inputs, thereby estimating the linear associative memory matrix. The inverse problem is thus handled by solving a series of forward problems, each forward problem related to training the identification scheme through the development of a memory matrix. The solution of forward problems are usually substantially easier to obtain and, as such, there is a great wealth of knowledge and insight available on the properties (e.g., uniqueness, existence, etc.) of such solutions for a number of different classes of systems. Thus the use of forward solution techniques to solve inverse problems appears to be an attractive route and may offer considerable insights which might not be obvious were one to attack the inverse problem using standard techniques.

We show that the concept of using an associative linear memory may be a powerful one in parameter identification as applied to structural and mechanical systems. The results that have been obtained are good considering the small amount of response data that has been used and the relatively few training vectors. They are indeed comparable to, if not far better than, those that might be obtained by using either the extended Kalman filter or the RPEM method [9]. In addition, the method appears to be capable of simultaneously identifying parameters that describe *both* the system and its excitation. As such, these results have far exceeded our expectations.

While this preliminary study shows encouraging results, the method described here opens up a series of questions dealing with:

1) The selection of the initial training set and its size; this study found that the associative memory matrix appears to be fairly sensitive to these initial trial vectors because they set the stage for further adaptive training.
2) The selection of the times at which response data is sampled; this study found that the results of the estimation procedure are sensitive to the times at which the data are gathered.
3) Improved schemes for adaptive training, and the provision of rigorous results on convergence of the parameter estimates; these are lacking at this time.
4) The selection of the size of the response vector; while we have shown (see Table 3) that in general increasing the size of the response vector usually provides improved results,

very large sized response vectors were found to give rise to numerical problems due to ill-conditioning.

5) The robustness of the method with respect to corruption of measurements by noise; while the method appears to be fairly robust in this regard (see [15] and Table 2), more computer experimentation is clearly called for.

We hope to address some of these issues in our future work.

REFERENCES

1. R.E. Kalaba and K. Spingarn, *Control, Identification and Input Optimization*, Plenum, (1982).
2. G. Rodriguez, Ed., *Proceedings of the Workshop on Identification and Control of Flexible Structures*, Vols. I and II, JPL Publication, 85-29 (April 1985).
3. P. Hajela and L. Berke, Neurobiological computations in structural analysis and design, *31st AIAA/ASME/ASCE Structures, Structural Dynamics and Materials Conference* (1990).
4. F.E. Udwadia and H. Flashner, Tradeoffs between identification and control in dynamic systems, *Journal of Applied Mechanics* **110**, 939–945 (1988).
5. F.E. Udwadia, Control of continuous systems, *Proceedings of the XV SECTAM Conference*, Atlanta, 1990.
6. J.J. Hopfield and D.W. Tank, Computing with neural circuits: A model, *Science* **233**, 625–633 (August 1986).
7. A.P. Sage and J.M. Melsa, *Estimation Theory with Applications to Communications and Control*, (McGraw-Hill 1971).
8. R.E. Kalaba and L. Tesfatsion, Solving nonlinear equations by adaptive homotopy continuation, *Applied Mathematics and Computation* (to appear).
9. L. Ljung, *System Identification: Theory for the User*, McGraw-Hill, (1988).
10. F.E. Udwadia, J. Garba and A. Ghodsi, Parameter identification problems, *Structural and Geotechnical Engineering* **110** (9), 1409–1432 (1984).
11. F.E. Udwadia and D.K. Sharma, Some uniqueness problems in the identification of building structural systems, *SIAM Journal of Applied Mathematics* **34**, 104–118 (1978).
12. T. Kohonen, *Self Organization and Associative Memory*, Springer-Verlag, New York, (1988).
13. B. Kosko, Bidirectional associative memories, *IEEE Trans. Sys., Man, and Cybern. SMC-18*, 49–60 (1988).
14. R.E. Kalaba, Z. Lichtenstein, T. Simchomy and L. Tesfatsion, Linear and nonlinear associative memories for parameter estimation, *Inform. Science.* (to appear).
15. R.E. Kalaba and L. Tesfatsion, Obtaining initial parameter estimates for nonlinear systems using multicriteria associative memories, MRG Working Paper No. M9006, University of Southern California, Los Angeles, (1990).
16. D. Rumelhart and J. McClelland, Eds., *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, Vol. 1 and 2, MIT Press, Cambridge, (1988).
17. S. Sharma and P. Hajela, Neural network applications in structural analysis, *Proceedings of the XV SECTAM Conference*, Atlanta, 1990.
18. D. Rehak, C.R. Thewalt and L.B. Doo, Neural network applications in structural mechanics computations, *Journal of Computer Utilization* (1989).
19. R. Penrose, On best approximate solutions of linear matrix equations, *Proc. Cambridge Phil. Society* **52**, 17–20 (1956).