NASA Technical Memorandum 104177

Neural Network Modeling of Nonlinear Systems Based on Volterra Series Extension of a Linear Model

Donald I. Soloway
NASA Langley Research Center

Jan T. Bialasiewicz
University of Colorado at Denver

January 1992

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, VA 23665

NEURAL NETWORK MODELING OF NONLINEAR SYSTEMS BASED ON VOLTERRA SERIES EXTENSION OF A LINEAR MODEL

Donald I. Soloway
NASA Langley Research Center
Automation Technology Branch, Information Systems Division
Mail Stop 152D, Hampton, VA   23665-5225

Jan T. Bialasiewicz
University of Colorado at Denver
Electrical Engineering Department
Campus Box 110, P.O. Box 173364, Denver, CO   80217-3364

ABSTRACT

A Volterra series approach has been applied to the identification of nonlinear systems which are described by a neural network model. A procedure is outlined by which a mathematical model can be developed from experimental data obtained from the network structure. Applications of the results to control of robotic systems are discussed.

INTRODUCTION

The Volterra series[1] approach to the identification of nonlinear systems which is presented in this paper, is a natural extension of earlier results in modeling of linear systems[2]. In reference 2, the impulse response of a linear dynamical system is shown to be given by the weights of the network model. For nonlinear systems, in addition to the impulse response of the linear approximation, the higher order Volterra kernels must be expressed in terms of the parameters of the trained network model. This relationship is reported in this paper. The result obtained means that it is possible not only to obtain a neural network model of the nonlinear dynamics, but also to represent this model by a mathematical expression. This opens a broad range of applications for the neural network modeling of nonlinear dynamical systems. The Volterra series in neural networks literature appeared recently in references 3 and 4. Both papers showed that a model of the Volterra system can model a nonlinear analytic system. However, this result follows directly from the representation theorem, proved in reference 5. Some interesting

results of neural network applications to nonlinear systems control can be found in reference 6.

In robotics, there are many places where nonlinear processes exist. The nonlinearities to be controlled include motor dynamics, flexible beam vibrations, harmonic drive stiffness, gear backlash, and full arm dynamics. Some of these nonlinearities, for example, beam vibrations and full arm dynamics, can be classified as analytic nonlinearities. This paper shows how to obtain a mathematical model of these nonlinearities using experimental data collected from the system under investigation.

In manipulator control, it is required that the manipulator respond quickly and accurately in spite of existing nonlinearities and inter-joint couplings. To obtain a good design, one should use as much a priori knowledge as possible and compliment the design with an adaptive fine tuning algorithm. In principle, this is the structure of the control scheme proposed by Koivo[7]. In this structure, shown in figure 1, the primary controller is developed based on the available model of the manipulator and the secondary controller compensates for unmodeled dynamics. Investigating the design of the primary controller is proposed, using a nonlinear model of the manipulator to be obtained as a Volterra series representation of a neural network model. The system fine tuning can be done, if necessary, by an adaptive loop using a Linear Quadratic Gaussian approach. In the proposed design, the model-based approach[8] and the performance-based approach[9] would be merged to obtain better performance.
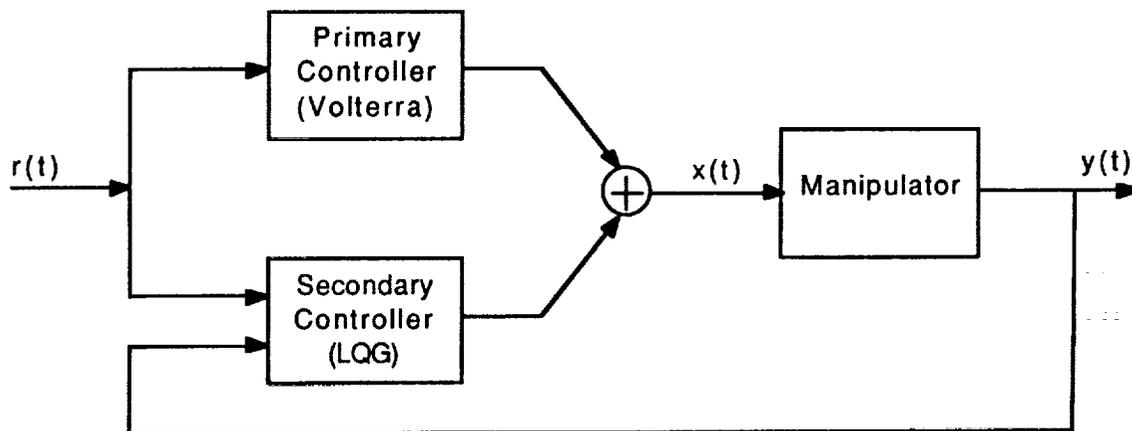

Figure 1.- Manipulator System

MODELING OF LINEAR SYSTEMS

The input-output relation of a system described by a linear differential equation may be given by a convolution integral

$$y(t)= \int_0^\infty h(\tau)x(t-\tau)d\tau \tag{1}$$

that specifies the output $y(t)$ in terms of the input $x(t)$ and the system impulse response $h(\tau)$. The discrete-time representation has the following form:

$$y(k)= \sum_{n=0}^\infty h(n)x(k-n) \tag{2}$$

where the arguments k and n are shorthand for kT and nT with T being the sampling interval to be selected for any particular system. Information on the system bandwidth of interest is needed to choose a proper value for T.

The relation (2) becomes approximate when a finite number of terms r is considered, that is, when

$$y(k)= \sum_{n=0}^{r-1} h(n)x(k-n) \tag{3}$$

which results in unmodeled dynamics, represented by the truncated terms. Equation (3), written using standard neural network notation, is

$$y(k)= \sum_{n=0}^{r-1} w_{1n}x(k-n) \tag{4}$$

that is, the finite (truncated) impulse response is given by:

$$[h(0) \ h(1) \cdots h(r-1)]=[w_{10} \ w_{11} \cdots w_{1r-1}] \tag{5}$$

This relationship, at any time instant k, can be viewed as a representation of a neural network with r inputs $x(k-i)$, $i=0,1,\cdots,r-1$ and a single output $y(k)$, generated by a single linear neuron. This network can be considered a member of the Σ^r class of feedforward networks[5].

Once r is fixed for a linear system, no modeling improvement can be reached by increasing the number of nodes and/or the number of layers. However, the increase of the number of nodes/layers will result in a structure redundancy and the robustness to neuron failure will be obtained. Consequently, the time needed to recover from a failure will be shorter.

The network model of a linear system, discussed so far, is shown in figure 2, in which q^{-1} denotes the unit delay operator, that is,

$$q^{-1}x(k)=x(k-1).$$

This network is described by the following difference equation:

$$(w_{10}+w_{11}q^{-1}+w_{12}q^{-2}+...+w_{1n}q^{-n})x(k)=y(k) \qquad (6)$$

which is equivalent to equation (4). Also, (6) can be represented as the vector product

$$y(k)=\Phi^T(k)\theta(k) \qquad (7)$$

with $\Phi^T(k)=[x(k)\ x(k-1)\ ...\ x(k-n)]$ and $\theta^T(k)=[w_{10}(k)w_{11}(k)\ ...\ w_{1n}(k)]$. It should be emphasized that by using the finite input sequence an approximate model (7) of the system (2), known as the Finite Impulse Response (FIR) model, is obtained.
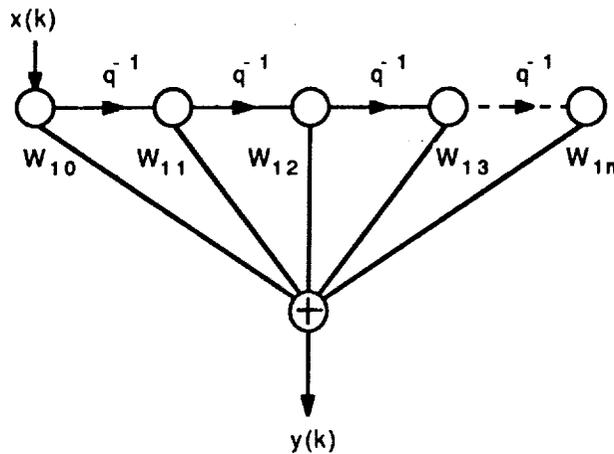

Figure 2.- Single node FIR network

MODELING OF NONLINEAR SYSTEMS

ANALYTIC SYSTEMS

Let a Single-Input, Single-Output (SISO) nonlinear dynamical system be described by a functional

$$y(t)=F[x(t)] \tag{8}$$

where $x(t)$ is the input, $y(t)$ is the output and the functional F is analytic, such that it can be represented exactly by a converging infinite series of the following form:

$$y(t)= \int_0^\infty h_1(\tau)x(t-\tau)d\tau + \int_0^\infty \int_0^\infty h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 + ...$$

$$+ \int_0^\infty ... \int_0^\infty h_n(\tau_1,...,\tau_n)x(t-\tau_1)x(t-\tau_2)...x(t-\tau_n)d\tau_1 d\tau_2...d\tau_n + ... \tag{9}$$

Such a system can be represented to any desired degree of accuracy by a finite series of the form of (9). This equation, known as the Volterra series expansion, can be interpreted as a functional generalization of the Taylor series expansion and represents the solution to a large class of nonlinear differential equations. For a linear system only the first term in expression (9) is nonzero and represents the convolution integral, with $h_1(\tau)$ being the impulse response of the system.

In expression (9), $h_n(\tau_1,...,\tau_n)$, $n=2,3,...$ are higher order Volterra kernels, or higher order impulse responses, introduced to describe nonlinear dynamic behavior. If (8) is discretized, then (9) assumes the form

$$y(k)= \sum_{n=1}^\infty y_n(k) \tag{10}$$

where

$$y_n(k)= \sum_{n_1=0}^\infty ... \sum_{n_n=0}^\infty h_n(n_1,n_2,...,n_n)x(k-n_1)...x(k-n_n) \tag{11}$$

ACTIVATION FUNCTION SUITABLE FOR MODELING OF NONLINEAR DYNAMICAL SYSTEMS

Let us assume that this equation (11) is to be modeled by a network. This implies the requirement that the number of inputs to the network

5

is finite. However, assume that the equation could be modeled by a Σ^r network with the network input defined as follows:
$$x^T=[x(k)\ x(k-1)\ ...\ x(k-r+1)] \qquad (12)$$
where r will be a number of time delayed inputs, and an activation function $\Psi(\cdot)$ as an operator which is applied to the sum of the weighted inputs to a node to produce an output from the node.

We have assumed so far that the modeling of nonlinear dynamical systems will require a multiple node Σ^r network.

Definition: The activation function for Σ^r networks, suitable for modeling nonlinear systems, is defined as a function $\Psi: R \rightarrow [a, b]$ which is differentiable, nondecreasing, $\lim\limits_{\lambda \rightarrow \infty} \Psi(\lambda)=b$ and $\lim\limits_{\lambda \rightarrow -\infty} \Psi(\lambda)=a$.

Two examples of an activation function, both widely used in Σ^r networks, are the logistics function and the hyperbolic tangent function:

1. Logistics function $\Psi: R \rightarrow [0, 1]$, defined by the following equation:

$$\Psi(\lambda)= \frac{1}{1+e^{-\lambda}} \qquad \text{with the derivative} \quad \Psi'(\lambda)=\Psi(\lambda)(1-\Psi(\lambda))$$

2. Hyperbolic tangent function $\Psi: R \rightarrow [-1, 1]$, defined by the following equation:

$$\Psi(\lambda)=\tanh(\lambda)= \frac{1-e^{-2\lambda}}{1+e^{-2\lambda}} \quad \text{with the derivative} \quad \Psi'(\lambda)=1-\Psi^2(\lambda)$$

Note that the computation of the derivatives of both functions, repeatedly performed while using a gradient method such as the error backpropagation for training, is computationally efficient.

According to the $\Psi(\cdot)$ definition, the following functions are not the activation functions :

1. $\qquad \Psi(\lambda)=\text{sign}\lambda$

2. $\qquad \Psi(\lambda)= \begin{cases} a & \text{for } \lambda < a/\alpha \\ \alpha\lambda & \text{for } a/\alpha < \lambda \leq b/\alpha \\ b & \text{for } \lambda > b/\alpha \end{cases}$

with a<b and α being a positive constant.

Now, let us justify the definition of $\Psi(\cdot)$ and the choice of one of those activation functions. First of all, let us realize that the activation function of a node should, for a given input x, encode within finite limits the amount of the preferred feature represented by this node and generated by this input. This preferred feature is abstracted in the learning process by the vector of weights w. The closeness of two vectors w and x is given in the vector space by their inner product

$$w \cdot x = \sum_{i=1}^{n} w_i x_i \qquad (13)$$

$$w \cdot x = \|w\| \, \|x\| \, \cos\theta \qquad (14)$$

where θ is the angle between the vectors w and x. By requiring the activation to be nondecreasing, the sense of closeness established by the inner product is preserved. Furthermore, since the derivative of the activation function is used in training the backpropogation network, the derivative must exist. Continuity is not sufficient.

Also, note that for both the logistics function and the hyperbolic tangent function the requirements formulated so far hold. However, the linear approximation of tanh($\cdot$) in the neighborhood of zero is the straight line passing through zero. It was shown in an earlier paper[2] that with such an activation function the product of the weight matrices of a multiple layer feedforward network, trained to model linear dynamics, represents the impulse response of a simulated linear system. Futhermore, it is shown below that if the single hidden layer network, using the hyperbolic tangent as an activation function, is trained to represent the nonlinear system (10), then its weights form the kernel $h_1(\cdot)$ of the functional $y_1(k)$ given by (11). The kernel $h_1(\cdot)$ is interpreted as the impulse response of the linear approximation of a nonlinear system. It is important to have a simple relation between $h_1(\cdot)$ and the network weights. Consequently, the hyperbolic tangent is a better choice of an activation function.

NETWORK REPRESENTATION OF SISO NONLINEAR SYSTEMS

Using the hyperbolic tangent as the activation function, a nonlinear system can be modeled by a delay network generating the vector x, defined by (12), and followed by a single hidden layer network $\Sigma^r(\Psi)$ with q nodes. In other words, the model is defined as

7

$$f: R^r \rightarrow R: \quad f(x) = \sum_{j=1}^{q} w_j \tanh(net_j), \quad w_j \in R, \qquad (15)$$

where

$$net_j = b_j + \sum_{i=1}^{r} w_{ji} x(k-i+1) \quad \text{and} \quad b_j \in R$$

Therefore, the dynamics of a complete network model are defined by the weight matrix $W=[w_{ji}]$, the output weight vector $w=[w_1 \ w_2 \ ... \ w_q]^T$, and the bias vector $b=[b_1 \ b_2 \ ... \ b_q]^T$. Hornik *et al.*[5] proved that the Σ^r network is capable of approximating any Borel measurable function from R^r to R to any desired degree of accuracy, provided sufficiently many hidden nodes are available. If the system to be modeled has the input-output relation $y=F(x)$, with the input vector $x=[x(k) \ x(k-1) \ ... \ x(k-r+1)]^T$, such that at any sampling instant k it represents a nonlinear Borel measurable function from R^r to R, the following claim follows directly from the representation theorem in reference 5.

Claim: Under the assumptions made above and for a fixed r, the accuracy of the approximation of a system modeled increases with an increase of the number q of available hidden nodes. This accuracy can be improved to any desired degree by increasing both r and q.

Note that under the assumptions made, the signal-dependent nonlinearities such as hysteresis are excluded.

Example of Obtaining Volterra Kernals

First, we shall partially analyze the network, showing the relations between the weights of a trained network and the Volterra kernels of the system modeled. Assume that the network is of the form shown in figure 2, with only one node (r=2) and no bias (b=0). Assuming that the nonlinear system (8) is analytic, then the network output $y^*(k)$ is given by:

$$y^*(k) = \tanh(w_{11}x(k) + w_{12}x(k-1)) \qquad (16)$$

Expanding $y^*(k)$ into a Volterra series, one obtains

8

$$y^*(k)=w_{11}x(k)+w_{12}x(k-1)-\frac{1}{3}w_{11}^3x^3(k)-w_{11}^2w_{12}x^2(k)x(k-1)$$

$$-w_{11}w_{12}^2x(k)x^2(k-1)-\frac{1}{3}w_{12}^3x^3(k-1)$$

$$+(\text{fifth-order terms})-(\text{seventh-order terms})+... \qquad (17)$$

If a nonzero bias is assumed, then the even-order terms will appear in the expansion (17). It was assumed that the system output $y(k)$ can be represented by a Volterra series, that is, equation (10) holds.

Assuming as above r=2 and b=0,

$$y(k)=\sum_{n=0}^{1}h_1(n)x(k-n)+\sum_{n=0}^{1}\sum_{m=0}^{1}h_2(n,m)x(k-n)x(k-m)$$

$$+\sum_{n=0}^{1}\sum_{m=0}^{1}\sum_{l=0}^{1}h_3(n,m,l)x(k-n)x(k-m)x(k-l)+... \qquad (18)$$

which, showing separately first-, second- and third-order terms, can be rewritten as

$$y(k)=h_1(0)x(k)+h_1(1)x(k-1)$$
$$+h_2(0,0)x^2(k)+[h_2(0,1)+h_2(1,0)]x(k)x(k-1)+h_2(1,1)x^2(k-1)$$
$$+h_3(0,0,0)x^3(k)$$
$$+[h_3(1,0,0)+h_3(0,1,0)+h_3(0,0,1)]x^2(k)x(k-1)$$
$$+[h_3(1,1,0)+h_3(1,0,1)+h_3(0,1,1)]x(k)x^2(k-1)$$
$$+h_3(1,1,1)x^3(k-1)+...$$

The coefficients in this expression are to be equated to the sum of proper coefficients expressed in terms of network weights in each of q expressions of the type (17). From this analysis, that is, assuming q=1, one can find that

$$h_1(0)=w_{11}$$
$$h_1(1)=w_{12}$$

.·

.

.

In other words, one can obtain Volterra kernels from the trained network. Some of those kernels will not be uniquely defined. Instead, their sum, e.g., $h_3(1,0,0)+h_3(0,1,0)+h_3(0,0,1)$ will be equal to a constant uniquely defined by the network parameters. In such a case, this constant can be arbitrarily distributed among the components of this

sum. For any distribution, the model obtained will have the same properties.

Using the network (15) for modeling, the output y(k) of a nonlinear system is, in general, approximated by

$$y^*(k)=\sum_{j=1}^{q}w_j\tanh(net_j) \tag{19}$$

If the function $\tanh(\cdot)$ is replaced by its Taylor series, then

$$y^*(k)=\sum_{j=1}^{q}w_j[net_j-\frac{1}{3}(net_j)^3+\frac{2}{15}(net_j)^5-...] \tag{20}$$

with $net_j=b_j+\sum_{i=1}^{r}w_{ji}x(k-i+1)$. In equation (20), the expression in brackets corresponds to the expression given by (17) for the specific example discussed above. On the other hand, according to (10),

$$y(k)=\sum_{n=0}^{r-1}h_1(n)x(k-n)+\sum_{n=0}^{r-1}\sum_{m=0}^{r-1}h_2(n,m)x(k-n)x(k-m)$$

$$+\sum_{n=0}^{r-1}\sum_{m=0}^{r-1}\sum_{l=0}^{r-1}h_3(n,m,l)x(k-n)x(k-m)x(k-l)+... \tag{21}$$

The coefficients in this expression are then equated to the coefficients in the expression for y*(k) given by equation (20). As a result one obtains the following equations specifying the first three Volterra kernels in terms of the parameters of the network model:

$$h_1(n)=\sum_{j=1}^{q}w_jw_{j\ n+1}(1-\tanh^2(b_j)),$$

$$n=0,...,r-1 \tag{22}$$

$$h_2(n,m)=\sum_{j=1}^{q}w_jw_{j\ n+1}w_{j\ m+1}(-2\tanh(b_j)+2\tanh^3(b_j))/2!,$$

$$n=0,...,r-1,\quad m=0,...,r-1 \tag{23}$$

$$h_3(n,m,l)=\sum_{j=1}^{q}w_jw_{j\ n+1}w_{j\ m+1}w_{j\ l+1}(-2+8\tanh^2(b_j)-6\tanh^4(b_j))/3!,$$

$$n=0,...,r-1,\quad m=0,...,r-1,\quad l=0,...,r-1 \tag{24}$$

If necessary, the equations specifying the higher order Volterra kernels can be obtained.

Concluding Remarks

This paper has demonstrated a method of determining the Volterra series representation of an analytic nonlinear dynamical system from a neural network that was trained on the nonlinear system to be identified. This procedure can be used to obtain a Volterra kernel with respect to a Taylor expansion of an arbitrary order. A simple example was demonstrated and the equations for the first three Volterra kernels were presented.

Current work is focused on a formal derivation of the general equations, in terms of network parameters, for any Volterra kernel.

REFERENCES

[1]   Schetzen, M.: "The Volterra and Wiener Theories of Nonlinear Systems", John Wiley and Sons, New York, 1980.

[2]   Bialasiewicz, J.T. and Soloway, D.: "Neural Network Modeling of Dynamical Systems", Proc. 5th IEEE Intl Symposium on Intelligent Control, Philadelphia, 1990, pp. 500-505.

[3]   Davis, G.W. and Gasperi, M.L.: "ANN Modeling of Volterra Systems," Proc. IJCNN, 1991, Vol.2, pp. 727-734.

[4]   Ahmed, H.M. and Rauf, F.: "NADINE-A Feedforward Neural Network for Arbitrary Nonlinear Time Series", Proc. IJCNN, 1991, Vol.2, pp. 721-726.

[5]   Hornik, K.; Stinchcombe, M. and White, H.: "Multilayer Feedforward Networks are Universal Approximators," Neural Networks, 1989, Vol.2, pp. 359-366.

[6]   Ydstie, B.E.: "Forecasting and Control Using Adaptive Connectionist Networks", Computers Chem. Eng., 1990, Vol. 14, No.4/5, pp. 583-599.

[7]   Koivo, A.J.: "Fundamentals for Control of Robotic Manipulators", John Wiley & Sons, Inc., New York, 1989.

[8]   Bejczy, A.K.: "Robot Arm Dynamics and Control", Tech. Memo. 33-669, Jet Propulsion Lab, Pasadena, 1974.

[9]   Dubowsky, S. and DesForges, D.T.: "The Application of Model-Referenced Adaptive Control to Robotic Manipulators", ASME J. Dyn. Syst. Meas. Contr., 1979, Vol. 101, No.3, pp. 193-200.

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE January 1992	3. REPORT TYPE AND DATES COVERED Technical Memorandum

4. TITLE AND SUBTITLE
Neural Network Modeling of Nonlinear Systems Based on Volterra Series Extension of a Linear Model

5. FUNDING NUMBERS
307-50-11-08

6. AUTHOR(S)
Donald I. Soloway and Jan T. Bialasiewicz

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23665-5225

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA TM 104177

11. SUPPLEMENTARY NOTES
Donald I. Soloway: Langley Research Center
Jan T. Bialasiewicz: University of Colorado at Denver

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified - Unlimited

Subject Category 63

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A Volterra series approach has been applied to the identification of nonlinear systems which are described by a neural network model. A procedure is outlined by which a mathematical model can be developed from experimental data obtained from the network structure. Applications of the results to control of robotic systems are discussed.

14. SUBJECT TERMS
Volterra
Linear Model   Neural Network

15. NUMBER OF PAGES
12

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		