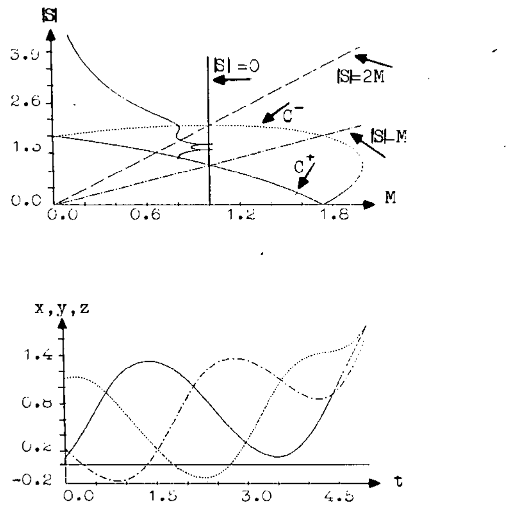

Fig. 5. A digitally obtained solution of (12). Initial conditions here are located (similarly to the initial conditions in the case of Fig. 4) in the region where either stability or instability are not *a priori* assured, namely: $M \leqslant |S| \leqslant 2M$. Initial conditions are: $S(0)=1.14$, $M(0)=0.80$, $\phi(0)=2$, this corresponds to $x_1(0)=0.07$, $x_2(0)=0.91$, $x_3(0)=0.16$. The oscillator parameters are the same as in Fig. 2.

this region and is not likely to impose "strange" dynamic features into the dynamic behavior of the modelled power system. This supports our earlier conclusions [1]–[3] based on simulations, which were also compared with experimentally observed dynamic behavior of real power systems.

The dynamic behavior in the instability region R^+ (21) is demonstrated in Fig. 3. Figs. 4 and 5 show a stable and an unstable solution, respectively, which originate at the same initial point $(|S(0)|, M(0))$ in R^0 (25), and differ only in their initial phases $\phi(0)$. These solutions indicate that there exist a sensitive dependence on the initial conditions in the region R^0, which is known as "chaotic behavior" [6].

We believe that the present work gives further support to a recently suggested [1]–[3] method of power systems representation. There has been a recently renewed interest in synchronization stability problems of power systems [7]. The relationship which exists between power systems analysis methods and methods which relate to circuits and systems theory has been emphasized [7]. The present work may be regarded as a further step in relating the field of circuit theory with methods of investigation which are applicable to the dynamics of power systems. The association of a power system behavior with a similar behavior of an oscillator may be even helpful in trying to study the problem of synchronization dynamics in power networks in the light of the relatively large literature on coupled oscillator systems [8].

The present work may also be regarded as a modest step in developing methods for analytically tackling nonlinear oscillators. It appears that the transformation to cyclotomic variable [5] possesses potentially useful applications in treating higher order oscillators [9]. These variables have also been used recently [10] to analyze a different type of three-phase oscillator of the general form

$$\dot{x}_k = \frac{\omega}{\sqrt{3}}(x_{k+1} - x_{k-1}) - \frac{\delta}{3}S + [\epsilon - \sigma(S, M)]x_k, \qquad k = 1,2,3 \pmod 3 \quad (30)$$

where in contrast to the σ_k given in (2), the stability function $\sigma = \sigma(S, M)$ in (30) is *independent* of the index k. The model (30) differs also from (1), in the fact that the asymptotic solutions of (30) are *independent* of the initial conditions $x_k(0)$, but depend only on the *sign* of the parameter δ.

References

[1] B. Z. Kaplan and D. Kottick, "Use of a three-phase oscillator model for the compact representation of synchronous generators," *IEEE Trans. Magn.*, vol. MAG-19, pp. 1480–1486, 1983.
[2] B. Z. Kaplan and D. Kottick, "A compact representation of synchronous motors and unregulated synchronous generators," *IEEE Trans. Magn.*, vol. MAG-21, pp. 2657–2663, 1985.
[3] D. Kottick and B. Z. Kaplan, "A compact representation of a two synchronous generators system," in *Proc. 1984 Summer Computer Simulation Conf.*, vol. 2, pp. 727–730. "Employment of three-phase compact oscillator models for representing comprehensively two synchronous generator systems," *Electric Machines and Power Syst.*, vol. 12, no. 5, 1987.
[4] B. Z. Kaplan and D. Kottick, "Another stabilized generator model for three-phase sine waves," *Math. Comp. Simulation*, vol. 23, pp. 277–279, 1981.
[5] J. Daboul, "Cyclotomic coordinates and their applications," in preparation.
[6] J. P. Eckmann and D. Ruelle, "Ergodic theory of chaos and strange attractors," *Rev. Mod. Phys.*, vol. 57, pp. 617–656, 1985.
[7] P. Varaiya, F. F. Wu, and R.-Liang Chen, "Direct methods for transient stability analysis of power systems: Recent results," *Proc. IEEE*, vol. 73, pp. 1703–1715, 1985.
[8] T. Endo and S. Mori, "Mode analysis of a ring of a large number of mutually coupled van der Pol oscillators," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 7–18, 1978.
[9] B. Z. Kaplan, "Development of conservative polyphase oscillators and some implications," *Int. J. Syst. Sci.*, vol. 14, pp. 409–416, 1983.
[10] J. Daboul and B. Z. Kaplan, "Solutions for a three-dimensional nonlinear stabilized oscillator," *Int. J. Circuit Theory Applications*, vol. 14, pp. 277–293, 1986.

Binary Petri-Net Relationships

H. ALAYAN AND R. W. NEWCOMB

***Abstract* —A full set of equations describing binary Petri-nets is presented in terms of integer algebra. In doing this, an equation is developed that allows for determination of the firing vector in terms of the markings and the input. Because the results are expressed in terms of integer arithmetic rather than Boolean algebra, they allow for extension to other classes of Petri-nets as well as yield ease of programming for Petri-net analysis.**

I. Introduction

Petri-nets are useful tools in analyzing and designing systems where discrete transitions occur, such as in clocked processors for assembly lines utilizing robots. In such cases, the actions may depend upon the presence or absence of just one item (or token-like signal) rather than many, while the classical theory of Petri-nets is set up in the many item framework. In addition, in the processor situation, when an output signal is present, the signal will get transmitted to all parts connected to it and able it to receive signals. In terms of Petri-nets, these properties mean that at most one token will be present in a place and that, when a transition fires, at the transition's output a token will appear on all places fed by the transition. Thus, we develop here the

Manuscript received April 30, 1986; revised October 2, 1986. The first author was supported by an AMIDEAST scholarship and the second author by the National Science Foundation under NSF Grant ECS-85-06924.
The authors are with the Microsystem Laboratory, Electrical Engineering Department, University of Maryland, College Park, MD 20742.
IEEE Log Number 8613479.

0098-4094/87/0500-0565$01.00

transition (or state-like) equations for timed Petri-nets where the number of tokens in a given place is a binary number and where every place connected to a fired transition receives a token. We call such a net a binary Petri-net. However, the development is within the domain of integer arithmetic, rather than Boolean arithmetic, so that the equations can be programmed in integer arithmetic and so that some of the techniques may be borrowed from (and some of the ideas may be carried over to) other than binary Petri-nets.

One of the primary features of Petri-nets is their nondeterministic nature, this being reflected by the fact that enabled transitions need not fire. Indeed the firing of enabled transitions is controlled by factors external to the Petri-net and must be read into the net by the user. This nondeterministic aspect makes Petri-nets difficult to use in computer routines and is avoided here by the use of inputs to places which in turn control the firing of otherwise enabled transitions. Making the inputs be controlled by external factors, rather than the firing of enabled transitions, allows for the development of a set of equations for the determination of the firing vector, and that is given here also. For Petri-net background, we refer to papers of Murata [1]–[3] and Peterson [4] as well as the book [5] of the latter.

II. Relationships

We recall that a Petri-net can be considered as a directed graph with two kinds of nodes: the places that hold tokens according to their markings, and the transitions that are enabled to fire according to the input places incident upon them. As illustrated by Fig. 1, the places are represented by circles, the tokens by dots in the places, and the transitions by bars; incident upon places are external inputs and transitions whereas incident upon transitions are solely places. Transitions and places may have many inputs and outputs. At the firing time λ, taken as discrete and normalized to a nonnegative integer, tokens are moved through a fired transition from places incident on the transition into the places on which the transition is incident to give the next marking for firing time $\lambda+1$. In the binary case, a marking is either 0 or 1 and if 1 the 1 is transferred through all the transitions connected to the place which fires. Markings can also come into the network from the outside through the external input I and leave as outputs Y to the outside world via the firing of output transitions. A transition is only enabled to fire if all the places incident upon it are filled with tokens (that is, have one token in the binary case).

We will assume, in contradistinction to other Petri-net theories, that an enabled transition will actually fire at the next firing time. If it is desired to control this firing by outside events, we add a place to the transition to be controlled and put an external input into this place. Then, when all other places are filled, the transition is essentially enabled, except it must wait for the external input to appear in order to actually fire the transition by fully enabling it.

For a Petri-net of p places and t transitions, we denote the p-vector containing the number of markings (that is, tokens) at time λ as $M(\lambda)$ and the t-vector of firings of the transitions as $F(\lambda)$, an entry 1 denoting a firing and an entry 0 denoting no firing at the firing time λ, with the inputs (into places) being denoted by the p-vector $I(\lambda)$; all of these vectors contain nonnegative integers which, in our case of binary Petri-nets, are restricted to being 1 or 0. The connections of the network are represented in the standard way [1] by the $t\times p$ connection matrix C where

$$C = C^{+} - C^{-} \tag{1}$$

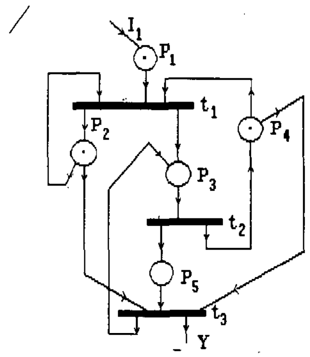

Fig. 1. Example binary Petri-net illustrating (4).

the entries c^{+}_{ij} of C^{+} being 1 or 0 depending upon whether transition i is incident upon place j or not and the entries c^{-}_{ij} of C^{-} being 1 or 0 depending upon whether place j is incident upon transition i or not.

The use of the connection matrix along with binary markings makes it convenient to work with both binary and normal integers in the same equations. Consequently, we introduce the notations $\doteq$ and $\square$ to allow us to carry out the necessary manipulations. Thus, the dot equality $\doteq$ means to carry out the operations on the right using normal integer arithmetic and in the final result to replace every positive number on the right by 1 and every other number by 0. The symbol $\square$ is first defined for a row p-vector a, having 0 and 1 entries, on a column p-vector b, having integer entries, to yield a binary scalar via

$$a\square b = \begin{cases} 1 & \text{if at least one of the pairs } a_i, b_i \text{ has} \\ & a_i = 1 \text{ and } b_i > 0 \text{ and no pair has } a_i = 1 \text{ and } b_i \le 0, \\ & \text{for } i = 1, \cdots, p \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

If in b we replace any positive entry by 1 and any nonpositive entry by 0, then $a\square b$ can be evaluated via the Boolean expression

$$(a_1' \vee b_1) \wedge (a_2' \vee b_2) \wedge \cdots \wedge (a_p' \vee b_p) \wedge \left[a_1 \wedge b_1 \vee a_2 \wedge b_2 \vee \cdots \vee a_p \wedge b_p \right] \tag{3}$$

where we use $\vee$, $\wedge$, and $'$ to denote logical OR, AND, and COMPLEMENT. The validity of (3) is seen by noting that if $a_i = 1$ and $b_i = 0$, then the term $(a_i' \vee b_i = 0)$ giving $a\square b = 0$, while if there are no such terms but at least one $a_i = 1$, in which case $b_1 = 1$, then the $(\cdot)$ terms will evaluate to 1 as does the $[\cdot]$ term. Having defined $\square$ on vectors, we next extend it to $t\times p$ matrices C on p-vectors M by using the above definition for the rows of C individually on M, that is, if the ith row of C is C_i then the ith entry of $C\square M$ is $C_i\square M$. In our use of $\square$, there will actually be negative terms to be considered (since a place with only one token can lose it through several simultaneously firing transitions to which it may be connected) and, thus, we have phrased the definition to handle arbitrary integers (or even real numbers if needed).

Following standard results [1] for Petri-nets, we know that if we were to allow more than one token in a place and there were no input, the marking at time λ would be $M_0(\lambda+1) = M(\lambda) + C^T F(\lambda+1)$, where the superscript T denotes matrix transposition. To reduce this marking vector to a vector which has only binary entries, we simply use the dotted equality introduced above. This makes these Petri-nets safe in the standard terminology [1, p. 45]. In order to take the input into account, we add it to the M_0 just found and again reduce the result using the dotted equality. To find the firing vector, we note that the ith transition actually fires, denoted by $f_i = 1$, when all the places incident upon the transition are filled with tokens (while $f_i = 0$ if it does not fire). By the way we have defined $\square$, this is denoted by $C^- \square M$. Since the output depends solely upon the components of the firing vector, we can also project F onto the output Y via an output (projection) matrix D. Writing these results together, we have

$$M(\lambda) \doteq M_0(\lambda) + I(\lambda) \tag{4a}$$

$$F(\lambda+1) = C^- \square M(\lambda) \tag{4b}$$

$$M_0(\lambda+1) \doteq M(\lambda) + C^T F(\lambda+1) \tag{4c}$$

$$Y(\lambda) = DF(\lambda). \tag{4d}$$

Equation (4a) gives the present marking in terms of the uninputted marking and the input, (4b) gives the next firing in terms of the present marking, while (4c) gives the next uninputted marking in terms of the present marking and the next firing. In conjunction with an initial marking, the state-like description of (4) represents a complete characterization of the Petri-net. The system is time-invariant in that we have assumed that the connection and output projection matrices are constant. We also have assumed that there is a delay in forming the firing vector from the present marking and that simultaneously the operation of taking the transition of the present marking into the next marking occurs. The systems described by (4) are also deterministic in that the present firing and next marking are completely determined by the present marking and input. This is because the nondeterministic behavior of the classical Petri-nets has been transferred to the input.

III. Examples

1) We illustrate (4) by the example Petri-net graphed in Fig. 1. Here

$$C = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & 1 \\ 0 & -1 & 1 & -1 & -1 \end{bmatrix}.$$

If we take the initial marking as $M_0(0) = [1,1,0,1,0]^T$ and the input constant $I(\lambda) = [1,0,0,0,0]^T$, we find

$$M(0) \doteq [M_0(0) + I(0)] \doteq [2,1,0,1,0]^T = [1,1,0,1,0]^T$$

$$F(1) = C^- \square M(0) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \square \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$M_0(1) \doteq M(0) + C^T F(0) \doteq \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$M(1) = [1,1,1,0,0]^T$$

and similarly

$$F(2) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, M_0(2) = M(2) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, F(3) = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

yielding

$$M_0(3) \doteq M(2) + C^T F(3) \doteq \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\doteq \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ 2 \\ -2 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$M(3) = [1,0,1,0,0]^T.$$

Continuing, we find

$$F(4) = F(2)$$

$$M(4) = [1,0,0,1,1]^T$$

and

$$F(5) = [0,0,0]^T.$$

Because $F(5) = 0$, the net has become deadlocked, because no further firings can occur, and, hence, for all $\lambda > 4$, $M(\lambda) = M(4)$ and $F(\lambda) = F(5)$. We also have the output equation as $Y(\lambda) = [0,0,1]F(\lambda)$. It should be noted that the system can be brought out of this deadlock by introducing an external input into place P_2.

It should be noted that our theory has both t_1 and t_3 firing simultaneously, which means that the customary meanings of liveness and safeness may not hold. For example, if one were to force t_1 and t_3 to fire at different times, then under the definition of liveness of [7, p. 68], and conversion to an equivalent marked graph, marking $M(2)$ is live since there are no token free loops. Similarly, under the definition L_4 of [8, p. 239], t_1 is live but not t_3 since one could obtain the perpetual firing of t_2, t_1. Of course, by the proper use of external inputs, one can control these firings and obtain the classical results also for the binary Petri-nets discussed here.

2) In Fig. 2, we present a binary Petri-net which could represent a production line of robots, the robots being represented by places $P_1 - P_4$ and the transitions $t_1 - t_5$ at their outputs. The choice of whether or not to fire an enabled transition is taken to be a choice made externally to the Petri-net. Thus, places $P_5 - P_9$ are inserted for this purpose with each of these places having a cooresponding input $I_5 - I_9$. It may be that place P_3 is to be considered as representing a backup robot that is called into play if a failure occurs in the robot represented by P_2. Thus, the choice of a token being routed through P_2 or P_3 is one made externally, say by a supervisor who decides that the direct line robot of P_2 should be replaced by the alternate path robot of P_3 so that repair can be undertaken of the former. Note that transitions t_1 and t_2 would normally require a decision to be

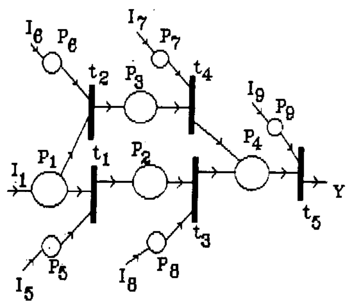

Fig. 2. Illustration of external control of enabled transitions.

made as to which one is to be fired when a token is present in P_1; this decision is made external to the Petri-net and carried out here by external inputs I_5 and I_6 (thus allowing us to describe Fig. 2 by (4)).

IV. Discussion

Here we have given Petri-net equations using ordinary integer arithmetic but valid for nets where the number of tokens are binary. To do this has necessitated some new operations, $\doteq$ and $\square$, in order to be able to express compactly the relationships as in (4). If one wishes to remain totally within a Boolean algebra framework, these operations can be further expressed via standard logic AND, OR, and COMPLEMENT operations as shown by (3). However, as also shown by (3), these expressions become rather messy, and, thus, when analyzing a binary Petri-net on a computer, it should be more convenient to stay within the full integer domain and use the expressions given here within that domain. We do comment that the dotted equality can be replaced by the $\square$, by noting the equivalence between $\doteq$ and $1_p\square$ (where 1_p is the $p \times p$ identity), in which case only one of these new operations, the square, needs to actually be implemented.

Equations (4) also hold for other than binary Petri-nets by suitably interpreting the two operations $\doteq$ and $\square$ introduced here. For example, if all the places get filled up when N tokens are in them, then the dotted equality can be used to replace by N any number bigger than N in the evaluation of the right of (4b).

The Petri-nets discussed here are timed with all delays normalized to be equal, though this is not a critical property for the ideas developed since the expressions developed can be used whenever a firing (simultaneously of all enabled transitions) is to take place. Further, we have made the systems deterministic so that the systems can be programmed for analysis on a computer. But again this is not a critical property, since, as discussed above, the important nondeterministic property of classical Petri-nets is contained via the use of external inputs to control the firing of otherwise enabled transitions. As one reviewer suggests, perhaps the next step is to consider inputs that are stochastic processes. And, as commented upon by another reviewer, the meaning of liveness and safeness needs investigation. Thus, we need alternatives to the facts that, in conventional marked graphs, a live marking guarantees the absence of deadlock and that a safe marking guarantees the absence of overflows [9, p. 1010]. Of course these classical properties can be preserved if desired by the use of external inputs to control the firing of otherwise enabled transitions. On the other hand, as commented by a reviewer, it may be possible to obtain systems equivalent to the binary Petri-nets discussed here by some means of imposing a constraint of only one token per place within previous theories. Such an investigation seems worthwhile since it could lead to an alternative to (4).

References

[1] T. Murata, "Modeling and analysis of concurrent systems," in *Handbook of Software Engineering*, C. R. Vick and C. V. Ramamoorthy, Eds. New York: Van Nostrand Reinhold, 1984, pp. 39–63.

[2] T. Murata, "Petri nets and their application, an introduction," in *Management and Office Information Systems*, L. K. Chong, Ed. New York: Plenum, 1984, pp. 351–368.

[3] T. Murata, "State equation, controllability, and maximal matchings of Petri nets," *IEEE Trans. Automat. Contr.*, vol. AC-22, pp. 412–416, June 1977.

[4] J. L. Peterson, *Petri-net Theory and the Modeling of Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[5] J. L. Peterson, "An introduction to Petri nets," in *Proc. Nat. Electron. Conf.*, vol. 32, Oct. 1978, pp. 144–148.

[6] H. Alayan and R. W. Newcomb, "Petri-net robot models for robot networks," in *Proc. 1986 IEEE Int. Symp. Circuits and Syst.* (San Jose, CA), May 1986, pp. 996–999.

[7] T. Murata and J. Y. Koh, "Reduction and expansion of live and safe marked graphs," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 68–70, Jan. 1980.

[8] J. L. Peterson, "Petri nets," *Computing Surveys*, vol. 9, no. 3, pp. 223–252, Sept. 1977.

[9] R. Johnsonbaugh and T. Murata, "Additional methods for reduction and expansion of marked graphs," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 1009–1014, Oct. 1981.

A Subsequence Approach to Interpolation Using the FFT

JOHN W. ADAMS, SENIOR MEMBER, IEEE

***Abstract*—A novel approach to using the FFT for interpolating discrete-time signals is presented. It is shown that the number of arithmetic operations can be reduced by decomposing the interpolated signal into an ordered set of subsequences. An example is included to demonstrate the effectiveness of this approach.**

I. Introduction

The objective of this paper is to show how the efficiency of the FFT method for the interpolation of discrete-time signals can be improved significantly. The approach relies on the concept of dividing the output signal into an ordered set of subsequences. Each of the required subsequences is computed separately and then interleaved to form the interpolated signal. By properly defining the subsequences, the unnecessary computations used for the conventional FFT interpolation method can be eliminated.

II. Conventional FFT-Based Approach to Interpolation

The conventional FFT-based method for interpolation is described in [1] and is reviewed briefly in this section. The signal to be interpolated is denoted $s(n)$ and it is assumed to have a duration of N samples, where N is usually an integer power of 2. The interpolation error may be reduced by multiplying $s(n)$ by a weighting function $w(n)$. The weighted data is defined as $x(n)$

Manuscript received May 23, 1986. A portion of this work was presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing on April 1986, in Tokyo, Japan.

The author is with the Electrical and Computer Engineering Department, California State University, Northridge, CA 91330, and with the Radar Systems Group, Hughes Aircraft Company, Los Angeles, CA 90009.

IEEE Log Number 8613480