USC-SIPI REPORT #121

Time and Lag Recursive Computation of Cumulants from a State Space Model

by

Ananthram Swami and Jerry M. Mendel

July 1988

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 404
Los Angeles, CA 90089-2564 U.S.A.

Contents

List of Figures

Abstract

Time and lag recursive algorithms for the computation of the cumulants of the state vector and the output process of a multiple-input multiple-output (MIMO) time-varying state-space model (SSM) are derived by using a Kronecker product representation for the cumulants of vector processes. The noise processes are not assumed to be stationary. Elegant expressions are obtained when the SSM is time-invariant and in observable form. For MIMO linear processes, closed-form expressions relating the output cumulants to the SSM parameters and to the impulse response matrices are established. Symmetry relations for the cumulants of vector processes are also discussed. Computational aspects are discussed in detail; and some system identification issues are addressed.

1. Introduction

During the past few years, *cumulants*, which are phase-sensitive higher-order statistics, have been used in a wide variety of signal processing and system theory problems, [5]-[9], [13], [15], [17], [19]-[26]. They are useful in the analysis and description of non-Gaussian processes, and non-minimum phase and non-linear systems. Current results in cumulant-based system modeling and identification are largely limited to stationary scalar processes and ARMA models (which include AR and MA models as special cases).

In this report, we focus our attention on the computation of the cumulants of the output of multiple-input multiple-output (MIMO) linear systems. The kth order cumulant of a p-th order vector process has been defined via a collection of p^{k-1} $(p \times p)$ matrices in [22] and [13]. We adopt and extend the Kronecker notation of [23] and [9] in which the k-th order cumulant is represented as a p^k-element vector. We develop a unified approach that handles non-stationary input processes and time-varying system parameters. Finally, when our linear system is described as a state-space model, we are able to exploit the system dynamics so as to obtain lag- and time-recursive expressions for the output cumulants. Although some of our derivations are for second-, third- and fourth-order cumulants, extensions to cumulants of other orders are straightforward. In practical applications, it is not likely that cumulants of order greater than four would be used.

In Section 2, we describe the assumptions for our MIMO state-space model (SSM). Section 3 provides some background on cumulants of scalar processes. In Section 4, we use Kronecker products to define cumulants of vector processes. Sections 5 deals with the computation of the state-cumulant vector. Section 6 deals with the computation of the output cumulant vector; some applications in system identification are also discussed. Computational details are discussed in Section 7. Simulation results are given in Section 8. Some useful results on Kronecker products are given in Appendix A. Proofs of most of the theorems that are stated in Sections 3-7 are given in Appendix B.

2. The Model

Our MIMO SSM is described by

$$x(n+1) = \Phi(n)x(n) + B(n)w(n) \quad (1)$$
$$y(n) = \Psi(n)x(n) + v(n) \quad (2)$$

where $x(n) \in \mathcal{R}^{n_x}$, $w(n) \in \mathcal{R}^{n_w}$ and $v(n), y(n) \in \mathcal{R}^{n_y}$.

Our modeling assumptions are:

(AS1) The input process, $w(n)$, is zero-mean and non-Gaussian. We assume that $w(n)$ is independent of $w(m)$, for $n \neq m$, but we do not assume that the components of $w(n)$ are mutually independent. We also assume that all relevant cumulants (and moments) of $w(n)$ are finite. Since $w(n)$ is non-Gaussian, there exists a finite $k > 2$ such that its k-th order cumulant is finite and non-zero.

(AS2) The output noise process, $v(n)$, is zero-mean and independent of $w(n)$.

(AS3) The system is causal and exponentially stable.

(AS4) Matrices $\Phi(n)$, $B(n)$ and $\Psi(n)$ and the input noise statistics are known.

(AS5) $E\{x(0)\} = 0$.

The assumptions $E\{w(n)\} = 0$ and $E\{x(0)\} = 0$ guarantee that $E\{x(k)\} = 0$, $\forall k$. If either of these assumptions is not met, and the means are known, it is a simple matter to remove the mean from the state vector $x(k)$, and then work with a new zero-mean state vector. The notation $\Phi(n)$ indicates time-dependence; we have departed from the standard, but more cumbersome, notation of $\Phi(n+1, n)$. Obviously, some of these assumptions, i.e., (AS4), will be relaxed when we consider system identification issues in Section 6.

Our SSM, therefore, describes a MIMO linear system with time-varying parameters, and non-stationary, non-Gaussian excitation. For $i \geq 0$, the solution of the state equation (1) is given by [14, p. 117]

$$x(n+i) = A(n+i, n)x(n) + \sum_{j=0}^{i-1} A(n+i, n+j+1)B(n+j)w(n+j) \quad (3)$$

where *the state-transition matrix*, $A(n+i, n)$, is defined (for $i > 0$) by

$$A(n+i, n) = \Phi(n+i-1)\Phi(n+i-2)\cdots\Phi(n) \quad (4)$$

with

$$A(n, n) = I \quad (5)$$

The following identities are easily established:

$$\begin{aligned}
A(n+1,n) &= \Phi(n) &(6)\\
A(n+i,n) &= A(n+i,n-1)\Phi(n) &(7)\\
A(n+i,n) &= \Phi(n+i-1)A(n+i-1,n), \quad i>0 &(8)\\
A(n,m) &= A(n,m+i)A(m+i,m), \quad m \le m+i \le n &(9)
\end{aligned}$$

Equation (3) may be re-written as

$$x(n) = \sum_{k=0}^{n-1} A(n,k+1)B(k)w(k) + A(n,0)x(0) \tag{10}$$

which expresses the state-vector as the response of a time-varying system excited by the input process, $u(n)$, plus the transient due to the initial state $x(0)$.

In the time-invariant case, we have $A(n+i,n) = \Phi^i$; and (3) becomes [16, eq. (2-17)]

$$x(n+i) = \Phi^i x(n) + \sum_{j=0}^{i-1} \Phi^{i-j-1} Bw(n+j) \tag{11}$$

3. Cumulants: Scalar Processes

Let $\mathbf{v} = [v_1, v_2, ..., v_k]^t$ and $\mathbf{z} = [z(t), z(t+t_1), ..., z(t+t_{k-1})]^t$. Then, the *k-th order cumulant* of the scalar random-process $z(t)$ is defined [21, p. 871] as the coefficient of $(v_1 v_2 ... v_k)$ in the Taylor expansion of the *cumulant-generating function*, also known as the second characteristic function,

$$K(\mathbf{v}) = \ln E\{\exp(\mathbf{v}^t \mathbf{z})\} \tag{12}$$

The k-th order cumulant of $z(t)$ is thus defined in terms of its joint moments of orders up to k. An equivalent definition in terms of partitions is given in [3]. For zero-mean scalar processes, the second-, third- and fourth-order cumulants are given by

$$\begin{aligned} C_{2y}(t;\tau) &= E\{y(t)y(t+\tau)\} & (13)\\ C_{3y}(t;t_1,t_2) &= E\{y(t)y(t+t_1)y(t+t_2)\} & (14) \end{aligned}$$

and

$$\begin{aligned} C_{4y}(t;t_1,t_2,t_3) &= E\{y(t)y(t+t_1)y(t+t_2)y(t+t_3)\} \\ &- E\{y(t)y(t+t_1)\}E\{y(t+t_2)y(t+t_3)\} \\ &- E\{y(t)y(t+t_2)\}E\{y(t+t_3)y(t+t_1)\} \\ &- E\{y(t)y(t+t_3)\}E\{y(t+t_1)y(t+t_2)\} \end{aligned} \tag{15}$$

Note that the fourth-order cumulant of a zero-mean random process equals its fourth-order moment less the fourth-order moment of a Gaussian random process with the same autocorrelation. Other properties of cumulants of stationary processes are discussed in [24, Ch II]. In accordance with (13), we will let $C_{2y}(t+t_1, t_2-t_1)$ denote $E\{y(t+t_1)y(t+t_2)\}$.

Brillinger and Rosenblatt [4] showed that the k-th order cumulant of the output, $y(n)$, of an exponentially stable SISO causal system, excited by an i.i.d. process, $w(n)$, is given by

$$C_{ky}(\tau_1, ..., \tau_{k-1}) = \gamma_{kw} \sum_{i=0}^{\infty} h(i)h(i+\tau_1)\cdots h(i+\tau_{k-1}) \tag{16}$$

where γ_{kw} is the k-th order cumulant of the input and $h(i)$ is the system impulse response (IR).

The focus of this report will be on the computation of the cumulants of the system output $y(n)$, both for stationary and non-stationary systems. In particular, our linear model in Section 2 is described as a state-space model with time-varying parameters and non-stationary inputs. One obvious method to compute the desired cumulant is to compute the system IR, given the SSM triple (i.e., $h(i) = \Psi\Phi^{i-1}B,\ i > 0$), and then use (16), provided the system is L.T.I. and stationary. In practice, the infinite summation would have to be truncated. Additionally, it seems redundant to compute the IR when the system's internal description is given. Further, we are interested in computing cumulants of time-varying, possibly MIMO, systems. Finally, since the state-variable model is recursive, and since second-order statistics can be computed recursively, we expect to develop recursive equations for the cumulants as well. In order to do this, we need to compute the cumulants of the state-vector. Hence, even in the SISO case, we need to define cumulants of vector processes.

4. Cumulants: Vector Processes

If $y(t)$ is a vector process of dimension, say p, i.e., $y(t) = [y_1(t), y_2(t), ..., y_p(t)]'$, we have two choices. We could let $v_i = [v_{i1}, ..., v_{ip}]'$, $i = 1, ..., k$, and $\mathbf{y} = [y'(t), y'(t + t_1), ..., y'(t + t_{k-1})]'$ in (12) (note that now, $\mathbf{v}$ and $\mathbf{y}$ in (12) are vectors of dimension pk). Alternatively, we could define the cross-cumulants of the elements of the vectors and then gather them into a single vector. Thus, if $y(t)$ is zero-mean, analogous to (13)-(15), we have

$$C_{y_i,y_j}(t;\tau) = E\{y_i(t)y_j(t+\tau)\} \tag{17}$$

$$C_{y_i,y_j,y_k}(t;t_1,t_2) = E\{y_i(t)y_j(t+t_1)y_k(t+t_2)\} \tag{18}$$

which is the third-order cross-cumulant of $(y_i(t), y_j(t+t_1), y_k(t+t_2))$. Further,

$$\begin{aligned} C_{y_i,y_j,y_k,y_l}(t;t_1,t_2,t_3) &= E\{y_i(t)y_j(t+t_1)y_k(t+t_2)y_l(t+t_3)\} \\ &- E\{y_i(t)y_j(t+t_1)\}E\{y_k(t+t_2)y_l(t+t_3)\} \\ &- E\{y_i(t)y_k(t+t_2)\}E\{y_l(t+t_3)y_j(t+t_1)\} \\ &- E\{y_i(t)y_l(t+t_3)\}E\{y_j(t+t_1)y_k(t+t_2)\} \end{aligned} \tag{19}$$

which is the fourth-order cross-cumulant of $(y_i(t), y_j(t+t_1), y_k(t+t_2), y_l(t+t_3))$, where $i, j, k, l = 1, ..., p$. Note that when $i = j = k = l$, (17)-(19) agree with (13)-(15).

Just as the cross-correlation of two random vectors is a collection of the cross-correlations of the scalar components of the random vectors, the (cross-)cumulants of random vectors are (ordered) collections of the cross-cumulants of the scalar components of the random vectors. If $x_1, ..., x_k$ are p-element random vectors, then, a natural representation for their cumulant, denoted by $\mathrm{cum}(x_1, ..., x_k)$ is a k-dimensional array, whose $(i_1, i_2, ..., i_k)$ element is $\mathrm{cum}(x_{1,i_1}, x_{2,i_2}, ..., x_{k,i_k})$, $i_1, ..., i_k = 1, ..., p$, where $x_{l,m}$ represents the m-th element of the vector x_l. Note that the scalar cross-cumulants are obtained via (12); see also (17)-(19). We will find it more convenient to represent the k-th order cumulant as a p^k element vector as defined below, but first we need some notation.

Notation. If $\mathbf{C}$ is a p^k element vector, we will let $\mathbf{C}[i_1, i_2, ..., i_k]$, $i_1, ..., i_k = 1, ..., p$, denote its $\big((i_1-1)p^{k-1} + (i_2-1)p^{k-2} + \cdots + i_k\big)$-th element. Essentially, the p^k-element vector is 'treated' as a k-dimensional array. Similarly, if M is a $p^k \times p^k$ matrix, we will represent its $\big((i_1-1)p^{k-1} + (i_2-1)p^{k-2} + \cdots + i_k, (j_1-1)p^{k-1} + (j_2-1)p^{k-2} + \cdots + j_k\big)$ element by $M([i_1, ..., i_k], [j_1, ..., j_k])$, $i_1, ..., i_k, j_1, ..., j_k = 1, ..., p$.

Definition. The cumulant of the p-element random vectors, $x_1, ..., x_k$, denoted by cum $(x_1, ..., x_k)$ is the p^k-element vector whose $[i_1, ..., i_k]$-element is given by $\mathrm{cum}(x_{1,i_1}, ..., x_{k,i_k})$, $i_1, ..., i_k = 1, ..., p$. In particular, the k-th order cumulant of the vector process $y(n)$, i.e., $\mathrm{cum}(y(n), y(n+\tau_1), ..., y(n+\tau_{k-1}))$, denoted by $\mathbf{C}_{ky}(n; \tau_1, ..., \tau_{k-1})$, has $\mathrm{cum}(y_{i_0}(n), y_{i_1}(n+\tau_1), ..., y_{i_{k-1}}(n+\tau_{k-1}))$, as its $[i_0, i_1, ..., i_{k-1}]$-th element, $i_0, ..., i_{k-1} = 1, ..., p$.

The motivation for representing cumulants as p^k-element vectors, rather than a k-dimensional array is two-fold: first, it enables us to use the usual algebra of vectors and matrices for computational purposes; second, it allows us to exploit the algebra of Kronecker products to obtain rather simple-looking expressions for the cumulants of vector processes.

The Kronecker product approach handles the MIMO case easily and leads to rather nice-looking formulae. We will let $\otimes$ denote the Kronecker product operator. A short review of Kronecker products and some useful results are given in Appendix A. The cumulants of vector processes are compactly expressed via Kronecker products:

Theorem 1: The k-th ($k = 2,3,4$) order cumulants of a zero-mean p-element vector process, $y(n)$, are given by the p^k vectors, $\mathbf{C}_{ky}$, as

$$\begin{aligned}
\mathbf{C}_{2y}(t;\tau) &= E\{y(t) \otimes y(t+\tau)\} && (20)\\
\mathbf{C}_{3y}(t;t_1,t_2) &= E\{y(t) \otimes y(t+t_1) \otimes y(t+t_2)\} && (21)\\
\mathbf{C}_{4y}(t;t_1;t_2,t_3) &= E\{y(t) \otimes y(t+t_1) \otimes y(t+t_2) \otimes y(t+t_3)\} \\
&- E\{y(t) \otimes y(t+t_1)\} \otimes E\{y(t+t_2) \otimes y(t+t_3)\} \\
&- P_p^T E\{y(t) \otimes y(t+t_2)\} \otimes E\{y(t+t_3) \otimes y(t+t_1)\} \\
&- P_p E\{y(t) \otimes y(t+t_3)\} \otimes E\{y(t+t_1) \otimes y(t+t_2)\} && (22)
\end{aligned}$$

where P_p is the $p^4 \times p^4$ permutation matrix given by

$$P_p = I \otimes U_{p^2 \times p} \tag{23}$$

in which the $(p^3 \times p^3)$ permutation matrix $U_{p^2 \times p}$ has unity entries in elements $[(i-1)p + k, (k-1)p^2 + i]$, $i = 1, ..., p^2$ and $k = 1, ..., p$, and zeros elsewhere. Matrix $U_{p^2 \times p}$ is formally defined in (104) in Appendix A.

Just as the correlation matrix of a vector process is an ordered collection of the cross- and auto-correlation terms of the scalar components of the vector process, the cumulants of vector processes should be ordered collections of the cross- and auto-cumulants of the scalar components of the vector process. We have used Kronecker products to obtain such a representation. Thus, rather than use $E\{y(t)y^T(t+\tau)\}$ for the usual $p \times p$ correlation matrix, we use the $p^2 \times 1$ vector $\mathbf{C}_{2y}$ in (20).

The cumulants of random vectors satisfy the following properties:

[CP1.] If Λ_i, $i = 1, ..., k$ are constant $r \times p$ matrices, and x_i, $i = 1, ..., k$ are p-element random vectors, then,

$$\text{cum}(\Lambda_1 x_1, ..., \Lambda_k x_k) = [\Lambda_1 \otimes \cdots \otimes \Lambda_k] \text{cum}(x_1, ..., x_k)$$

[CP2.] The cumulant vector is symmetric in its arguments in the following sense:

$$\text{cum}(x_1, ..., x_k) = U_{p^{k-j} \times p^j} \text{cum}(x_{j+1}, ..., x_k, x_1, ..., x_j)$$

where the U matrices are defined in (104) in Appendix A.

[CP3.] Cumulants are additive in their arguments, i.e.,

$$\operatorname{cum}(x_0 + y_0, z_1, ..., z_k) = \operatorname{cum}(x_0, z_1, ..., z_k) + \operatorname{cum}(y_0, z_1, ..., z_k)$$

[CP4.] If α is a constant vector, then

$$\operatorname{cum}(\alpha + z_1, ..., z_k) = \operatorname{cum}(z_1, ..., z_k)$$

[CP5.] If the random vectors $\{x_i\}_{i=1}^k$ are independent of the random vectors $\{y_i\}_{i=1}^k$, then,

$$\operatorname{cum}(x_1 + y_1, ..., x_k + y_k) = \operatorname{cum}(x_1, ..., x_k) + \operatorname{cum}(y_1, ..., y_k)$$

[CP6.] If a subset of the random vectors $\{x_i\}_{i=1}^k$ is independent of the rest, then

$$\operatorname{cum}(x_1, ..., x_k) = 0$$

Properties [CP1] through [CP6] are generalizations of the corresponding properties of the cumulants of random variables, [24, Ch II.2]. Properties [CP3]-[CP6] are obvious and are independent of the particular representation we have chosen in our definition above. Properties [CP1] and [CP2] are proved in Appendix B.

When the process is stationary, we will drop the time index t in $C_{ky}(t; ...)$ (see Theorem 1).

4.1 Equivalent Representations

Following classical second-order theory, the k-th ($k = 3, 4$) order cumulants of vector processes have been defined as a collection of p^{k-2} $(p \times p)$ matrices in [22] and [13] (stationary processes only) and as $p^2 \times p$ ($k = 3$) and $p^2 \times p^2$ ($k = 4$) matrices in [19].

In [22], the third-order cumulant is defined as the collection of p $(p \times p)$ matrices

$$\mathbf{C}^R_{3y,m}(t_1, t_2) = E\{y(t)y^T(t + t_1)Y_m(t + t_2)\}, \quad m = 1, .., p \tag{24}$$

where $Y_m(t) = diag(y_m(t), ..., y_m(t))$.

In [13] the third- and fourth-order cumulants are defined via

$$\mathbf{C}^{IGM}_{3y,m}(t_1, t_2) = E\{y(t + t_1)y^T(t)y_m(t + t_2)\}, m = 1, ..., p \tag{25}$$

$$\begin{aligned} \mathbf{C}^{IGM}_{4y,mn}(t_1, t_2, t_3) &= E\{y(t + t_1)y^T(t)y_m(t + t_2)y_n(t + t_3)\} \\ &- E\{y(t + t_1)y^T(t)\}E\{y_m(t + t_2)y_n(t + t_3)\} \\ &- E\{y(t + t_1)y_m(t + t_2)\}E\{y^T(t)y_n(t + t_3)\} \\ &- E\{y(t + t_1)y_n(t + t_3)\}E\{y^T(t)y_m(t + t_2)\} \\ &\quad m, n = 1, ..., p \end{aligned} \tag{26}$$

The cumulant matrix defined in (25) is the transpose of that defined in (24).

In [19], the ***diagonal*** slice of the third-order cumulant (i.e., $t_1 = t_2 = \tau$ in (18)) is defined as

$$\mathbf{C}_{3y,m}^{MW1}(t;\tau) = E\{y(t+\tau)y^T(t+\tau)y_m(t)\}, \quad m = 1, ..., p \tag{27}$$

and in mixed notation as the $p^2 \times p$ matrix

$$\mathbf{C}_{3y}^{MW}(t;\tau) = E\{y(t) \otimes [y(t+\tau)y^T(t+\tau)]\} \tag{28}$$

Similarly, the ***diagonal*** slice of the fourth-order cumulant (i.e., $t_1 = t_2 = t_3 = \tau$ in (19)) is defined as the $p^2 \times p^2$ matrix

$$\begin{aligned}\mathbf{C}_{4y}^{MW}(t;\tau) &= E\{y(t)y^T(t+\tau) \otimes y(t+\tau)y^T(t+\tau)\} \\ &- E\{y(t)y^T(t+\tau)\} \otimes E\{y(t+\tau)y^T(t+\tau)\} \\ &- E\{y(t) \otimes y(t+\tau)\}E\{y(t+\tau) \otimes y(t+\tau)\}^T \\ &- E\{[y(t) \otimes I]E\{y^T(t+\tau) \otimes y(t+\tau)\}[I \otimes y^T(t+\tau)]\}\end{aligned} \tag{29}$$

We note that the term $y^T(t+\tau) \otimes y(t+\tau)$ in (29) is the same as $y(t+\tau)y^T(t+\tau)$ since $y(t)$ is a column vector.

Let $\mathbf{C}_{3y}^{MW1}$ denote the $p^2 \times p$ matrix obtained by stacking the matrices $\mathbf{C}_{3y,m}^{MW1}$, $m = 1, ..., p$ defined in (27). Let $\mathbf{C}_{3y}^{IGM}$ and $\mathbf{C}_{3y}^{R}$ be similarly defined. Then, using the **vec** operator, which transforms matrices to vectors via column stacking [see Appendix A], we obtain

$$\mathbf{vec}\ [\mathbf{C}_{3y}^{MW1}(t;\tau)] = \mathbf{C}_{3y}(t+\tau; -\tau, 0) \tag{30}$$

$$\mathbf{vec}\ [\mathbf{C}_{3y}^{MW}(t;\tau)] = \mathbf{C}_{3y}(t+\tau; -\tau, 0) \tag{31}$$

$$\mathbf{vec}\ [\mathbf{C}_{4y}^{MW}(t;\tau)] = \mathbf{C}_{4y}(t+\tau; 0, -\tau, 0) \tag{32}$$

For stationary processes, we have

$$\mathbf{vec}\ [\mathbf{C}_{3y}^{MW1}(\tau)] = \mathbf{C}_{3y}(-\tau, 0) \tag{33}$$

$$\mathbf{vec}\ [\mathbf{C}_{3y}^{MW}(\tau)] = \mathbf{C}_{3y}(-\tau, 0) \tag{34}$$

$$\mathbf{vec}\ [\mathbf{C}_{4y}^{MW}(\tau)] = \mathbf{C}_{4y}(0, -\tau, 0) \tag{35}$$

$$\mathbf{vec}\ [\mathbf{C}_{3y}^{IGM}(t_1, t_2)] = \mathbf{C}_{3y}(t_2, t_1) \tag{36}$$

$$\mathbf{vec}\ [\mathbf{C}_{4y}^{IGM}(t_1, t_2, t_3)] = \mathbf{C}_{4y}(t_3, t_2, t_1) \tag{37}$$

The **vec** operator thus provides a convenient mechanism for converting the definitions in [22], [13] and [19] to the definitions given in (21) and (22). We *define* an $\mathbf{unvec}_{m,n}$ operator which converts an mn-element vector into an $m \times n$ matrix, such that

$$\mathbf{unvec}_{m,n}[\mathbf{x}] = X_{m \times n} \Longrightarrow \mathbf{vec}\ [X_{m \times n}] = \mathbf{x} \tag{38}$$

We have thus established a one-to-one correspondence between our Kronecker product representation in Theorem 1, and the various representations in [22], [13] and [19]. The definitions in Theorem 1 are much more compact than those given in (26) and (29). Further, there is a nice uniformity in the definitions in Theorem 1. Finally, note that Theorem 1, unlike the definitions given in [19], is not restricted to the 1-D diagonal slice.

4.2 Linear Vector Processes

Let us now consider a linear vector process $y(n)$, i.e.,

$$y(n) = \sum_{k=-\infty}^{\infty} H(n,k)w(k) \tag{39}$$

where $y(n) \in \mathcal{R}^{n_y}$, $w(n) \in \mathcal{R}^{n_w}$ and $H(n,k) \in \mathcal{R}^{n_y} \times \mathcal{R}^{n_w}$. We further assume that $w(n)$ is independent of $w(m)$, $n \neq m$; hence, its cumulants are multidimensional Kronecker delta functions, i.e.,

$$\mathbf{C}_{kw}(n;\tau_1,...,\tau_{k-1}) = \Gamma_{kw}(n)\ \delta(\tau_1)\cdots\delta(\tau_{k-1}) \tag{40}$$

where the input cumulant Γ_{kw} is an n_w^k-element vector. We also assume that the IR matrices $H(n,k)$ are absolutely summable so that the output cumulants are well-defined. Then, the counterpart of Brillinger and Rosenblatt's result in (16) is given by the following:

Theorem 2. For the linear vector process in (39), with $w(n)$ satisfying (40), the k-th order output cumulant is given by

$$C_{ky}(n;\tau_1,...,\tau_{k-1}) = \sum_{i=-\infty}^{\infty} [H(n,i) \otimes H(n+\tau_1,i) \otimes \cdots \otimes H(n+\tau_{k-1},i)]\Gamma_{kw}(i) \tag{41}$$

and in the stationary time-invariant case, when $H(n,k) = H(n-k)$ and $\Gamma_{kw}(n) = \Gamma_{kw}$, by

$$\mathbf{C}_{ky}(\tau_1,...,\tau_{k-1}) = \sum_{i=-\infty}^{\infty} [H(i) \otimes H(i+\tau_1) \otimes \cdots \otimes H(i+\tau_{k-1})]\Gamma_{kw} \tag{42}$$

Proof. See Appendix B.

Since the Kronecker products of scalars are themselves scalar products, we see that Brillinger-Rosenblatt's result given in (16) is the special case of (42) for SISO causal linear processes.

5. Cumulants of the State Vector

Using the Kronecker product approach, we will derive lag- and time-recursive equations for the k-th order cumulants of the state and output processes. For zero-mean processes, the second- and third-order cumulants are identical with the second- and third-order moments, and as such, are easily handled. The fourth-order cumulant, however, involves not only the fourth-order moments, but also the second-order moments, and in this sense, it is typical of higher-order cumulants. However, the results in Theorem 2 enable us to uniformly handle cumulants of all orders. Our development handles non-stationary processes and time-varying state-space models and leads to recursive (in lag and time) equations. Steady-state solutions are provided for the stationary time-invariant case. Recursive in time and lag equations for the *diagonal* slice of the third- and fourth-order cumulant of the state vector of an SSM are derived in [19] using a mixture of Kronecker product and matrix notations.

5.1 Time-Varying/Non-Stationary Case

In this section, we provide various algorithms for computing the k-th order cumulants of the state vector in (1). Closed-form expressions, as well as time- and lag-recursive equations are derived. Proofs of all theorems are given in Appendix B.

Theorem 3. Kronecker state-cumulant vector, $\mathbf{C}_{kx}(n; i_1, ..., i_{k-1})$, can be expressed, $\forall i_l$, $l = 1, ..., k-1$, in terms of its zero-lag values as follows:

$$\mathbf{C}_{kx}(n; i_1, ..., i_{k-1}) = [A(n, n-m) \otimes A(n+i_1, n-m) \otimes \cdots \otimes A(n+i_{k-1}, n-m)] \mathbf{C}_{kx}(n-m; 0, ..., 0) \quad (43)$$

where $m = -\min(0, i_1, ..., i_{k-1})$.

Theorem 3 handles lags in the entire $(k-1)$-D space, $-\infty < i_l < \infty$, $l = 1, ..., k-1$. Note that for causal systems $\mathbf{C}_{kx}(l; 0, ..., 0) = 0$ for $l < 0$. Further, with m as defined in Theorem 3, one of the A matrices in (43) is always the identity matrix, because $A(n + i_l, n + i_l) = I$; see (5).

Note that cumulants at positive lags ($i_l \geq 0$, $l = 1, ..., k-1$: $m = 0$) at time n are expressed in terms of the zero-lag value at time n; however, cumulants elsewhere at time n are expressed in terms of the zero-lag values at previous times $n - m$, $m > 0$.

Theorem 4. Kronecker state-cumulant vector, $\mathbf{C}_{kx}(n; i_1, ..., i_{k-1})$, can be computed recursively in its lag variables, i_l, $l = 1, ..., k-1$, for $0 \leq m_l \leq i_l$, as follows:

$$\begin{aligned} &\mathbf{C}_{kx}(n; i_1, ..., i_{k-1}) \\ &\quad = [I \otimes A(n + i_1, n + i_1 - m_1) \otimes \cdots \otimes A(n + i_{k-1}, n + i_{k-1} - m_{k-1})] \\ &\quad \mathbf{C}_{kx}(n; i_1 - m_1, ..., i_{k-1} - m_{k-1}) \end{aligned} \quad (44)$$

Thus, by setting $m_1 = 1$ and $m_j = 0$, $j = 2, ..., k-1$, we get a one-step recursion in the lag variable i_1.

Theorem 5. Kronecker state-cumulant vector, $\mathbf{C}_{kx}(n; i_1, ..., i_{k-1})$, can be computed recursively in its temporal variable, n, as follows, for $i_l \geq 0$, $l = 1, ..., k-1$:

$$\mathbf{C}_{kx}(n+1; 0, ..., 0) = \Phi_k(n)\mathbf{C}_{kx}(n; 0, ..., 0) + B_k(n)\Gamma_{kw}(n) \tag{45}$$

$$\begin{aligned}\mathbf{C}_{kx}&(n+1; i_1, ..., i_{k-1}) \\ &= [\Phi(n) \otimes \Phi(n+i_1) \otimes \cdots \otimes \Phi(n+i_{k-1})]\mathbf{C}_{kx}(n; i_1, ..., i_{k-1}) + D(n; i_1, ..., i_{k-1})\end{aligned} \tag{46}$$

where

$$D(n; i_1, ..., i_{k-1}) = [I \otimes A(n+i_1+1, n+1) \otimes \cdots \otimes A(n+i_{k-1}+1, n+1)]B_k(n)\Gamma_{kw}(n) \tag{47}$$

and for any matrix M, M_k is defined by

$$M_k = M_{k-1} \otimes M \tag{48}$$

with $M_1 = M$.

Although (45) is a special case of (46), we have chosen to include it because, as we shall soon see, the zero-lag cumulant plays a rather important role in the computation of cumulants.

Some comments are now in order:

1. The 1-D diagonal slice is obtained when $i_l = i$, $l = 1, ..., k-1$, in $\mathbf{C}_{kx}(n; i_1, ..., i_{k-1})$. The 1-D diagonal slice of third- and fourth-order cumulants of SISO processes have been successfully used in several signal processing applications, see [8] and [20], for example. In this case (i.e., $i_l = i$), (43) simplifies to

$$\mathbf{C}_{kx}(n; i, ..., i) = [I \otimes A_{k-1}(n+i, n)]\mathbf{C}_{kx}(n; 0, ..., 0), \quad i \geq 0 \tag{49}$$

$$\mathbf{C}_{kx}(n; i, ..., i) = [A(n, n+i) \otimes I_{k-1}]\mathbf{C}_{kx}(n+i; 0, ..., 0), \quad i \leq 0 \tag{50}$$

2. 1-D slices occur when $k-2$ of the $k-1$ lag variables are fixed, e.g., $i_l = i_1 + m_l$, $l = 2, ..., k-1$, where the m_l's are fixed, or when $i_l = m_l$, $l = 2, ..., k-1$, are fixed and i_1 is arbitrary. Some simplifications may be expected in this case; for example, (43), yields, for $i, m_l \geq 0$,

$$\begin{aligned}\mathbf{C}_{kx}&(n; i, i+m_2, ..., i+m_{k-1}) \\ &= [I \otimes A(n+i, n) \otimes A(n+i+m_2, n) \otimes \cdots \otimes A(n+i+m_{k-1}, n)]\mathbf{C}_{kx}(n; 0, ..., 0) \\ &= [I \otimes I \otimes A(n+i+m_2, n+i) \otimes \cdots \otimes A(n+i+m_{k-1}, n+i)]\mathbf{C}_{kx}(n; i, ..., i)\end{aligned} \tag{51}$$

   where we have used (43), (9) and property [P2] from Appendix A.

3. Equation (43) expresses all the cumulant lags explicitly in terms of the zero-lag cumulants. Equation (45) provides a time-recursion for the zero-lag cumulant; we assume that $\mathbf{C}_{kx}(0; 0, ..., 0)$ is given. Equation (44) is recursive in one or more cumulant lags; again, the zero-lag terms are required to initialize the recursion. Finally, for fixed lags, (46) provides recursion in time.

4. Equations (44)-(46) provide recursions in lag and time for the cumulant of the state vector; however, these equations are valid only for non-negative lags ($i_l \geq 0$). Corollary 1 (below) shows that cumulants at negative lags at time n can be obtained from cumulants at positive lags at times prior to n. Theorem 6 (below) gives a simultaneous time and lag recursion for negative lags.

Corollary 1. Kronecker state-cumulant vector, $\mathbf{C}_{kx}(n; i_1, ..., i_{k-1})$, where at least one of the (i_l)'s is negative, can be expressed in terms of cumulants at positive lags and earlier times, as follows:

$$\begin{aligned} &\mathbf{C}_{kx}(n; i_1, ..., i_{k-1}) \\ &= U_{p^{k-j} \times p^j} \mathbf{C}_{kx}(n + i_j; \overbrace{i_{j+1} - i_j, ..., i_{k-1} - i_j}^{k-j-1 \text{terms}}, -i_j, i_1 - i_j, ..., i_{j-1} - i_j) \\ &\qquad i_j \leq \min(0, \underbrace{i_1, ..., i_{j-1}}_{j-1 \text{terms}}, i_{j+1}, ..., i_{k-1}) \end{aligned} \tag{52}$$

where $p = dim(x(n))$, $j = 0, 1, .., k$, and the U matrices are defined in (104) in Appendix A.

Note that Corollary 1 naturally divides the $k-1$-D space into k regions of the form:

$$\begin{aligned} R_0 : &\quad 0 \leq \min(i_1, i_2, ..., i_{k-1}) \\ R_1 : &\quad i_1 \leq \min(0, i_2, ..., i_{k-1}) \\ R_2 : &\quad i_2 \leq \min(0, i_1, i_3, ..., i_{k-1}) \\ \cdots &\quad \cdots \\ R_{k-1} : &\quad i_{k-1} \leq \min(0, i_1, ..., i_{k-2}) \end{aligned} \tag{53}$$

Thus, in the case of third-order cumulants, we will find it convenient to divide the (i, j) plane into three segments: R_0 which covers QP I ($i, j \geq 0$); R_1 which covers QP II and the part of QP III on and above the $i = j$ line ($i \leq \min(0, j)$); and R_2 which covers the part of QP III below the $i = j$ line and QP IV ($j < \min(0, i)$), (see Figure 1). Note that according to (53), the origin is contained in all the R_l's; further, for $k = 3$, the line, $i = j \leq 0$ is contained in both R_1 and R_2. In order to avoid duplication, we have included the origin only in R_0 and the line, $i = j < 0$ only in R_1.

Since the system is causal, the Kronecker state-cumulant vector $\mathbf{C}_{kx}(n; i_1, ..., i_{k-1})$ is non-zero only when $n \geq 0$, $n + i_l \geq 0$, $l = 1, ..., k-1$. Consequently, given cumulant values in R_0 for times $0 \leq n < N$, all the non-zero cumulant values in regions $R_1, ..., R_{k-1}$ can be generated for all $n < N$. This is an important observation, since it considerably decreases the amount of computation required to evaluate the cumulants at negative lags; however, this requires storage of cumulant values in R_0.

Equation (43) provides explicit expressions for the cumulants at negative lags. By combining these with the time-recursion equation for the zero-lag term (45) and using (9), recursive equations (in lag and time) can also be derived in the same manner as for the positive lags.

Theorem 6. The Kronecker state-cumulant matrix, $\mathbf{C}_{kx}(n; i_1, \ldots, i_{k-1})$, for lags inside R_l, i.e., $i_l < \min(0, \underbrace{i_1, \ldots, i_{l-1}}_{l-1 \text{ terms}}, i_{l+1}, \ldots, i_{k-1})$, can be computed recursively in both its temporal variable, n, and its lag variable, i_l, as follows:

$$\begin{aligned}
&\mathbf{C}_{kx}(n; i_1, \ldots, i_l, \ldots, i_{k-1}) \\
&\quad = [\Phi(n-1) \otimes \Phi(n-1+i_1) \otimes \cdots \otimes \Phi(n-1+i_{l-1}) \otimes I \otimes \Phi(n-1+i_{l+1}) \\
&\quad\quad \otimes \cdots \otimes \Phi(n-1+i_{k-1})]\, \mathbf{C}_{kx}(n-1; i_1, \ldots, i_{l-1}, i_l+1, i_{l+1}, \ldots, i_{k-1}) \qquad (54)
\end{aligned}$$

Equation (54) is simultaneously recursive in time and in the negative lag i_l. Equations that are recursive only in time or in the negative lags can be obtained, but these will involve inversion of $\Phi(n)$. For example, for lags inside region R_{k-1}, where $i_{k-1} < 0$,

$$\begin{aligned}
\mathbf{C}_{kx}(n; i_1, \ldots, i_{k-1}) &= [I_{k-1} \otimes \Phi^{-1}(n+i_{k-1})]\mathbf{C}_{kx}(n; i_1, \ldots, i_{k-2}, i_{k-1}+1) \\
&- [\{\otimes_{l=0}^{k-1} A(n+i_l, n+i_{k-1}+1)\}]B_k(n+i_{k-1})\Gamma_{kw}(n+i_{k-1}) \qquad (55)
\end{aligned}$$

This equation is derived in Appendix B.

Theorems 3-6 and Corollary 1 provide several ways of recursively computing the state-cumulant vector in lag and in time. We discuss four schemes for the computation of the third-order cumulant, stressing the inherent parallelism that should make such schemes particularly amenable to parallel and systolic implementations:

- **Scheme 1.** (See Fig. 2) The zero-lag cumulant is computed recursively in time using (45) in Theorem 5. At each time point, cumulants in R_0 are computed via lag-recursion, Eq. (44) in Theorem 4. Cumulants in R_1 and R_2 are obtained via (52) of Corollary 1. This scheme requires the storage of cumulant values in R_0, but does not require any computations to obtain the cumulants in R_1 and R_2. This scheme permits parallel computation of the cumulants in R_0 at several time points.

- **Scheme 2.** At time $n = 0$, the lag-recursion, Eq. (44), in Theorem 4 is used to generate all desired lags in R_0. The time-recursion equations, (45) and (46) in Theorem 5, are then used to obtain, perhaps simultaneously in parallel, all desired cumulants in R_0 at the next time point. Cumulants in R_1 and R_2 are obtained via Corollary 1, as in Scheme 1. This scheme permits parallel computation of several cumulants in R_0, from one time point to the next.

- **Scheme 3.** Same as Scheme 1, except that cumulants in R_1 and R_2 are obtained via the combined time-and-lag recursion equations given in Theorem 6. This scheme does not require storage of the cumulants in R_0. Rather than use Theorem 6, one could use the pure lag recursion equations, such as (55); this would permit parallel computations in R_0, R_1 and R_2. However, this would also require inversion of the Φ matrix; additionally, (55) is more complicated than (54).

- **Scheme 4.** Same as Scheme 2, except that cumulants in R_1 and R_2 are obtained via the combined time-and-lag recursion equations given in Theorem 6. This scheme does not require storage of the cumulants in R_0. Rather than use Theorem 6, one could use the pure time recursion equations (counterparts of (55)); this would permit parallel computations in R_0, R_1 and R_2. However, this would also require inversion of the Φ matrix; additionally, (55) is more complicated than (54).

- If only 1-D slices of the third-order cumulant are required, then further savings in the above schemes may be possible, for example, by appropriately using equations (49)-(51).

As usual, we have a tradeoff between computational complexity and storage. The method in Scheme 1 is depicted in Fig. 2. Several observations are in order (similar observations, modulo appropriate changes, may be made for the other schemes):

1. All the cumulants in R_0 can always be computed; if $\mathbf{C}_{3x}(n; 0, 0)$ reaches steady-state (SS), all the cumulants in R_0 also reach SS. This is a consequence of causality.

2. For $n = 0$, all the cumulants in R_1 and R_2 are zero; this is a consequence of causality (hence, $x(n) = 0$, $n < 0$).

3. R_1 fills up from right to left as the temporal variable n increases. From (52), we note that at time n, R_1 has exactly n non-zero columns because $n + i \geq 0$ and $i < 0$. [More generally, from (52), we note that for our causal SSM, cumulants in R_j are defined only for $n + i_j \geq 0$ and $i_j < 0$]. As n gets larger, the entire R_1 region gets filled up; and when $\mathbf{C}_{3x}(n; 0, 0)$ reaches SS, so do all the cumulants in R_1.

4. R_2 fills up from top to bottom, as n increases. It has only n non-zero rows, because $n + j \geq 0$ and $j < 0$ [see (52) and the preceding item]. As n gets larger, the entire R_2 region gets filled up, and when $\mathbf{C}_{3x}(n; 0, 0)$ attains SS, so do all the cumulants in R_2.

5. R_1 and R_2 computations are non-recursive in a single variable. Cumulants in R_1 and R_2 use earlier values of cumulants in R_0.

6. Additional parallel processing schemes are possible in R_0, R_1 and R_2.

7. For the computation of k-th order cumulants, the $(k-1)$-D space is naturally divided into k regions (see Corollary 1); hence, observations similar to those made for the third-order cumulant would follow. Because of the extra dimensions, there is more parallelism in computing the higher order cumulants than there is in computing the third-order cumulants. Figure 2 would have $k+1$ rows, one for the zero-lag cumulant, and one for each of the R_l's, $l = 0, ..., k-1$.

5.2 Stationary, Time-Invariant Case

When $\Phi(n) = \Phi$, $A(n+i,n) = \Phi^i$. Several simplifications occur in this case, e.g., (43) becomes

$$\begin{aligned} \mathbf{C}_{kx}(n; i_1, ..., i_{k-1}) &= [\Phi^m \otimes \Phi^{i_1+m} \otimes \cdots \otimes \Phi^{i_{k-1}+m}]\mathbf{C}_{kx}(n-m; 0, ..., 0) && (56) \\ &= [I \otimes \Phi^{i_1} \otimes \cdots \otimes \Phi^{i_{k-1}}]\Phi_k^m \mathbf{C}_{kx}(n-m; 0, ..., 0) && (57) \end{aligned}$$

where $m = -\min(0, i_1, ..., i_{k-1})$; and, in Theorem 5,

$$D(n; i_1, ..., i_{k-1}) = [I \otimes \Phi^{i_1} \otimes \cdots \otimes \Phi^{i_{k-1}}]B_k(n)\Gamma_{kw}(n) \tag{58}$$

In the **stationary, time-invariant** case, [i.e., when $\Phi(n) = \Phi$, $B(n) = B$ and $\Gamma_{kw}(n) = \Gamma_{kw}$], when the state vector is in steady-state, the temporal index, n, can be dropped from the cumulant notation $\mathbf{C}_{kx}(n; i_1, ..., i_{k-1})$, i.e., $\mathbf{C}_{3x}(n; i, j) \longrightarrow \mathbf{C}_{3x}(i, j)$.

Corollary 2. Steady-state Kronecker state-cumulant vector, $\mathbf{C}_{kx}(i_1, ..., i_{k-1})$, can be expressed in terms of its zero-lag values $\forall\, i_l$, $l = 1, ..., k-1$, as follows:

$$\mathbf{C}_{kx}(i_1, ..., i_{k-1}) = [\Phi^m \otimes \Phi^{i_1+m} \otimes \cdots \otimes \Phi^{i_{k-1}+m}]\ \mathbf{C}_{kx}(0, ..., 0) \tag{59}$$

where $m = -\min(0, i_1, ..., i_{k-1})$; and the zero-lag term, $\mathbf{C}_{kx}(0, ..., 0)$, is given by

$$\mathbf{C}_{kx}(0, ..., 0) = [I_k - \Phi_k]^{-1} B_k \Gamma_{kw} \tag{60}$$

Proof. Equation (59) follows directly from Theorem 3, and (60) follows from (45) in Theorem 5. □

Equation (59), for $k = 3$, with $i_1, i_2 \geq 0$, is also given in [7].

In steady-state, the Kronecker state-cumulant vector has a lot of inherent symmetry. Recall, [24], that in the stationary scalar case, we have

$$\begin{aligned} C_{3x}(i,j) = {} & C_{3x}(j,i) && = C_{3x}(-j, i-j) = \\ C_{3x}(i-j, -j) = {} & C_{3x}(j-i, -i) && = C_{3x}(-i, j-i) \end{aligned} \tag{61}$$

Hence, all the cumulants in the plane can be computed from the cumulants in the wedge $\{(i,j) : i \geq 0, j \leq i\}$ (see Figure 3a). This is true in the vector case as well, as shown in:

Lemma 1. For a stationary vector process, the Kronecker state-cumulant vector, $\mathbf{C}_{3x}(i,j)$, in the wedge, $\{(i,j) : i \geq 0, j \leq i\}$, specifies the cumulant everywhere on the plane, via

$$\begin{aligned} \mathbf{C}_{3x}(i,j) &= [I \otimes U_{p \times p}]\mathbf{C}_{3x}(j,i) \\ &= U_{p^2 \times p}\mathbf{C}_{3x}(-j, i-j) \\ &= U_{p^2 \times p}[I \otimes U_{p \times p}]\mathbf{C}_{3x}(i-j, -j) \\ &= U^T_{p^2 \times p}\mathbf{C}_{3x}(j-i, -i) \\ &= U^T_{p^2 \times p}[I \otimes U_{p \times p}]\mathbf{C}_{3x}(-i, j-i) \end{aligned} \tag{62}$$

where the $p^2 \times p^2$ matrix $U_{p\times p}$ and the $p^3 \times p^3$ matrix $U_{p^2\times p}$ are defined in (104) in Appendix A.

Proof. See Appendix B.

More generally, the symmetry relations for k-th order cumulants are given by

$$\begin{aligned} &\mathbf{C}_{kx}(i_1, ..., i_{k-1}) \\ &\quad = U_{p^{k-j}\times p^j}\mathbf{C}_{kx}(i_{j+1} - i_j, ..., i_{k-1} - i_j, -i_j, i_1 - i_j, ..., i_{j-1} - i_j) \end{aligned} \tag{63}$$

where $j = 0, ..., k$. Equation (63) follows directly from Corollary 1; it does not express *all* of the inherent symmetries (compare (63) with (62), with $k = 3$); however, other symmetry relations are easily derived.

Note that once the zero-lag SS values are obtained, SS cumulant values at all other lags can be computed from (59) and symmetry relations such as those given in Lemma 1 for third-order cumulants. The $p^k \times p^k$ matrix $(I_k - \Phi_k)$ is guaranteed to be non-singular because of the assumed stability of our SSM. Standard inversion techniques would require on the order of p^{3k} *flops*. An alternative to explicitly inverting $(I_k - \Phi_k)$ is to use the following iterative procedure:

Theorem 7. Iterating the equation

$$\mathbf{C}_{kx}(n+1; 0, ..., 0) = \Phi_k \mathbf{C}_{kx}(n; 0, ..., 0) + B_k \Gamma_{kw} \tag{64}$$

with respect to the variable n leads to the steady-state solution given in (60) of Corollary 2, for *any* finite-valued initial condition, provided Φ has all of its eigenvalues within the unit circle.

Proof. This result follows directly from [P4] in Appendix A. Equation (64) describes a state-equation with step excitation; if the system is stable, steady-state is attainable. □

The number of iterations required for convergence of the algorithm given in Theorem 7 depends upon the eigenvalue distribution of Φ.

The matrix Φ^k is a $p^k \times p^k$ matrix; thus direct evaluation of (64) would involve p^{2k} units of storage; further, the computational complexity is $O(p^{2k})$. In Section 7, we present computational schemes that reduce the storage to p^k elements, and the computational complexity to $O(p^{k+1})$.

Yet another alternative to explicit inversion is to recast (60) as a matrix Lyapunov equation. Let $A = B = \Phi_{k/2}$, for k even, and let $A = \Phi_{(k+1)/2}$ and $B = \Phi_{(k-1)/2}$, for k odd. Then, using the **vec** and **unvec** operators, we can rewrite

$$\mathbf{y} = [I - A \otimes B]^{-1}\mathbf{x}$$

as the discrete Lyapunov matrix equation

$$Y - BYA^T = X \ ,$$

which can be solved in $O(p^{2k})$ flops.

If the matrix Φ is in one of the standard canonical forms, it has only $(2p-1)$ non-zero blocks, of which $p-1$ are identity matrices. In this case, matrix Φ_k has only $(2p-1)^k$ non-zero elements, of which $(p-1)^k$ blocks are identity matrices. Hence, sparse matrix techniques [10] may be used for the matrix inversion.

If the matrix Φ is non-defective (i.e., its Jordan form is strictly diagonal) then, we may exploit Kronecker product property [P4] as follows. Let $\Phi = V\Lambda V^{-1}$ denote the eigenvector decomposition of Φ, where Λ is strictly diagonal. Then, using [P4] and [P5], we obtain

$$(I_k - \Phi_k)^{-1} = V_k(I_k - \Lambda_k)^{-1}(V^{-1})_k$$

Thus, when Φ is non-defective, the effort required to compute the inverse in (60) is essentially that required for inverting the $p \times p$ matrix V. Further, if the matrix V is unitary (e.g., Φ has distinct eigenvalues), then, the explicit inversion of V is not required $(V^{-1} = V^H)$. When the matrix Φ is defective, either the iterative procedure given in Theorem 7, or the discrete Lyapunov equation method must be used. See, also Section 7.

6. Cumulants of the Output Vector

Given the SSM triple $(\Phi(n), B(n), \Psi(n))$, the impulse response matrices of the system can be computed as $H(n,k) = \Psi(n)A(n,k+1)B(n)$, $n > 0$ [see eqs. (10) and (4)], and in the time-invariant case as $H(n) = \Psi\Phi^{n-1}B$, $n > 0$. The output cumulants can then be computed using the infinite summation in Theorem 2, (42). We will derive closed-form expressions for the output cumulants directly in terms of the model parameters; this will, incidentally, yield a closed-form expression for the summation in (42), and, hence, for (16).

6.1 Non-recursive Equations

In this section we will give closed-form expressions for the output cumulants in terms of the SSM matrices.

6.1.1 Obvious Results

Theorem 8. The k-th order Kronecker output-cumulant vectors can be calculated as,

$$\mathbf{C}_{ky}(n;\tau_1,...,\tau_{k-1}) = [\Psi(n) \otimes \Psi(n+\tau_1) \otimes ... \otimes \Psi(n+\tau_{k-1})]\mathbf{C}_{kx}(n;\tau_1,...,\tau_{k-1}) \tag{65}$$

and in the steady-state, stationary, time-invariant case, as,

$$\mathbf{C}_{ky}(\tau_1,...,\tau_{k-1}) = \Psi_k \mathbf{C}_{kx}(\tau_1,...,\tau_{k-1}) \tag{66}$$

Proof. Follows directly from (2) and property [CP1]. □

Equation (66), for $k = 3$ is also given in [7].

Since we already have closed-form expressions for the Kronecker state-cumulant vector, it is easy to compute the output cumulants using Theorem 8.

In the stationary, time-invariant case, Giannakis [5] notes that for SISO models the positive lags of the cumulant slice $c_{3y}(m, n_0)$, $m > 0$, n_0 fixed, can be obtained as the impulse response of the SSM (Φ, g, ψ), where Φ and ψ are identical to those in the original SSM (Φ, b, ψ), and g is expressed in terms of an infinite summation involving the impulse response. Using Kronecker products, we can extend this result to arbitrary cumulant orders and to MIMO models; further, we obtain closed-form expressions for the g matrix. Since the k-th order cumulant is a function of $(k-1)$ lag variables, we have to freeze $k-2$ of the lag variables. Our results are given in:

Theorem 9. For the model in (1) and (2), with SSM triple, (Φ, B, Ψ), the Kronecker output cumulant vector can be expressed, in the stationary time-invariant case, as the IR of the SSM triple (Φ, G, Ψ), as follows, for $\tau_l \geq 0$, $l = 1, ..., k-2$, and $\tau_{k-1} > 0$:

$$\mathbf{C}_{ky}(\tau_1,...,\tau_{k-1}) \;=\; [I_{k-1} \otimes \Psi\Phi^{\tau_{k-1}-1}]G_{k;\tau_1,...,\tau_{k-2}} \tag{67}$$

where

$$G_{k;\tau_1,\ldots,\tau_{k-2}} = [\Psi \otimes \Psi\Phi^{\tau_1} \otimes \ldots \otimes \Psi\Phi^{\tau_{k-2}} \otimes \Phi]\mathbf{C}_{kx}(0,\ldots,0) \tag{68}$$

$$= [\Psi_{k-1} \otimes I]\mathbf{C}_{kx}(\tau_1,\ldots,\tau_{k-2},1) \tag{69}$$

and

$$\mathbf{C}_{kx}(0,\ldots,0) = [I_k - \Phi_k]^{-1}B_k\Gamma_{kw} \tag{70}$$

The I matrix in (67) is $(n_y \times n_y)$, whereas that in (69) is $(n_x \times n_x)$. For single-output models, I_{k-1} in (67) is the unit scalar.
Further,

$$\bar{C}_{ky}(\tau_1,\ldots,\tau_{k-1}) = \Psi\Phi^{\tau_{k-1}-1}\bar{G}_{k;\tau_1,\ldots,\tau_{k-2}} \tag{71}$$

where

$$\bar{G}_{k;\tau_1,\ldots,\tau_{k-2}} = \mathbf{unvec}_{n_x,n_y^{k-1}}[G_{k;\tau_1,\ldots,\tau_{k-2}}] = \bar{C}_{kx}(\tau_1,\ldots,\tau_{k-2},1)\Psi_{k-1}^T \tag{72}$$

$$\bar{C}_{ky}(\tau_1,\ldots,\tau_{k-1}) = \mathbf{unvec}_{n_y,n_y^{k-1}}[C_{ky}(\tau_1,\ldots,\tau_{k-1})] \tag{73}$$

and

$$\bar{C}_{kx}(\tau_1,\ldots,\tau_{k-2},1) = \mathbf{unvec}_{n_x,n_x^{k-1}}[\mathbf{C}_{kx}(\tau_1,\ldots,\tau_{k-2},1)] \tag{74}$$

Proof. Follows directly be applying the lag-variable recursion theorem to Theorem 8; details are given in Appendix B. Equation (71) follows immediately from (67) by using the definitions of the **vec** and **unvec** operators, which are given in Appendix A. □

Although Theorem 9 holds only for positive lags, similar expressions for other lags may be obtained by making use of symmetry relations, such as those given in Lemma 1 for the third-order cumulant. Equation (71) would seem to indicate that the two SSM triples, (Φ, B, Ψ) and $(\Phi, \bar{G}, \Psi)$ have the same observability space; however, this need not be true, as is easily established in the SISO case.

Theorems 8 and 9 give closed-form expressions for the output cumulants in terms of the SSM parameters; they also provide closed-form expressions for the Brillinger-Rosenblatt result given in (16), and its MIMO extension in (42).

In the SISO case, the equations in Theorem 9 simplify to

$$C_{ky}(\tau,\ldots,\tau_{k-1}) = \psi\Phi^{\tau_{k-1}-1}g_{k;\tau_1,\ldots,\tau_{k-2}}$$

where

$$g_{k;\tau_1,\ldots,\tau_{k-2}} = [\psi_{k-1} \otimes I]\mathbf{C}_{kx}(\tau_1,\ldots,\tau_{k-2},1)$$

6.1.2 A Less Obvious Result: Stationary Time-Invariant Case

Although we have time- and lag-recursive equations for the state-cumulants, the equations in Theorems 8 and 9 do not lead to recursive equations for the output process, since the matrices Ψ and Φ will not, in general, be commutative. In this subsection, we will assume that the SSM matrices are time-invariant, and that the noise processes are stationary. From Lemma 1, and its generalization, the k-th order cumulants of a stationary linear process are specified everywhere, via symmetry relationships, by the cumulant lags, $C_{ky}(i_1, ..., i_{k-1})$ in the non-redundant 'wedge' $0 \leq i_1 \leq i_2 \leq ... \leq i_{k-1}$; see Fig. 3a for the $k = 3$ case.

When the stationary process is a linear process, generated by a p-th order (p = dim(Φ)) SSM, the non-redundant volume is given by $0 \leq i_1 \leq i_2 \leq ... \leq i_{k-1} < p$; see Fig. 3b for the $k = 3$ case; cumulants elsewhere in the 'wedge', and hence elsewhere in the volume, may be found via recursive equations as we show below.

Let $\alpha(\lambda) = det(\lambda I - \Phi)^{-1} = \sum_{m=0}^{p} \alpha_m \lambda^{p-m}$, $\alpha_0 = 1$, denote the characteristic polynomial of Φ; by the Cayley-Hamilton theorem, we have $\alpha(\Phi) = 0$. Hence, for $i, j, k \geq 0$, using (59) for $k = 4$, we find,

$$\begin{aligned}
0 &= \Psi_4[I \otimes \Phi^i \otimes \Phi^j \otimes \Phi^k \alpha(\Phi)]\mathbf{C}_{4x}(0,0,0) \\
&= \Psi_4 \sum_{m=0}^{p} \alpha_m [I \otimes \Phi^i \otimes \Phi^j \otimes \Phi^k \Phi^{p-m}]\mathbf{C}_{4x}(0,0,0) \\
&= \Psi_4 \sum_{m=0}^{p} \alpha_m \mathbf{C}_{4x}(i,j,k+p-m) \\
&= \sum_{m=0}^{p} \alpha_m \mathbf{C}_{4y}(i,j,k+p-m) \qquad (75)
\end{aligned}$$

where we have also used (66) from Theorem 8. Consequently,

$$\mathbf{C}_{4y}(i,j,p+k) = -\sum_{m=1}^{p} \alpha_m \mathbf{C}_{4y}(i,j,k+p-m), \quad k \geq 0$$

Given $\mathbf{C}_{4y}(i,j,k), 0 \leq i \leq j \leq k < p$, all other cumulant values can, therefore, be computed using the above recursion and symmetry properties. Although we have demonstrated this only for fourth-order cumulants, extensions to other orders is obvious.

It would, therefore, be useful to develop an algorithm that provides all the cumulants in the non-redundant region in one shot. We will now develop such an algorithm.

From (2) and (3), we obtain

$$y(n+m) = \Psi(n+m)A(n+m,n)x(n) + \Psi(n+m)\sum_{j=0}^{m-1} A(n+m,n+j+1)B(n+j)w(n+j) + v(n+m) \,. \tag{76}$$

Let $Y^T(n) = [y^t(n), y^t(n+1), ..., y^T(n+p-1)]$, $W^T(n) = [w^t(n), w^t(n+1), ..., w^T(n+p-1)]$, and $V^T(n) = [v^t(n), v^t(n+1), ..., v^T(n+p-1)]$. Concatenating (76), for $m = 0, ..., p-1$, where $p = dim(\Phi)$, we obtain

$$Y(n) = \begin{bmatrix} \Psi(n)A(n,n) \\ \Psi(n+1)A(n+1,n) \\ \vdots \\ \Psi(n+p-1)A(n+p-1,n) \end{bmatrix} x(n) + T(n)W(n) + V(n) \tag{77}$$

which we will compactly represent as

$$Y(n) = O(n)x(n) + T(n)W(n) + V(n) \tag{78}$$

where the matrix $T(n)$ is block lower-triangular with the (m, j) block given by $\Psi(n + m)A(n+m, n+j+1)B(n+j)$, $j = 0, ..., m-1$ and $m = 0, ..., p-1$; matrix $T(n)$ has zeros on the main diagonal.

We will now compute the cumulants of the vector process $Y(n)$. Note that the random vectors $x(n)$, $W(n)$ and $V(n)$ on the right-hand side of (78) are mutually independent. The zero-lag k-th order cumulant of each of the terms on the right-hand side of (78) is readily computed via Theorem 2. From the way $Y(n)$ has been defined, **the zero-lag cumulants of $Y(n)$ yield the cumulants of $y(n)$ at all lags in the non-redundant region of support.**

Using properties [P1] and [P2], causality of the SSM, the definitions of cumulants of vector processes, and properties [CP1] and [CP5], we obtain from (78)

$$\mathbf{C}_{kY}(n) = O_k(n)\mathbf{C}_{kx}(n) + T_k(n)\mathbf{C}_{kW}(n) + \mathbf{C}_{kV}(n) \tag{79}$$

where $\mathbf{C}_{kY}$, $\mathbf{C}_{kx}$, $\mathbf{C}_{kW}$ and $\mathbf{C}_{kV}$ denote the zero-lag cumulants of the vector processes $Y(n)$, $x(n)$, $W(n)$ and $V(n)$. In the time-invariant case, we have $O(n) = O$, the observability matrix for the SSM triple, $\{\Phi, B, \Psi\}$, and $T(n) = T$, where T is a lower-triangular block-Toeplitz matrix, with (m, j) block given by $\Psi\Phi^{m-j-1}B = H(m-j)$, where the $H(n)$'s are the $n_y \times n_w$ impulse response matrices for our SSM. The (i, j) element of $H(n)$ is the response of the i-th output channel to impulse excitation at the jth input channel. In the SS stationary case, we obtain:

Theorem 10. The k-th order SS output cumulant vector $\mathbf{C}_{kY}$ for a time-invariant, stationary SSM is given by

$$\mathbf{C}_{kY} = O_k[I_k - \Phi_k]^{-1}B_k\Gamma_{kw} + T_k\mathbf{C}_{kW} + \mathbf{C}_{kV} \tag{80}$$

Proof. The SS solution to (79) follows directly by using (60). □.

The steady-state solution in (80), for $k = 3$ was given in [9] for SISO ARMA processes. Equation (80) was also derived in [23] for the fourth-order cumulant of SISO AR processes, with the SSM in observable form.

Theorem 10 gives us the SS zero-lag cumulant of the process $Y(n)$; from the definition of $Y(n)$ we see that $\mathbf{C}_{kY}$ does, in fact, give us all the cumulants of the process $y(n)$ in the non-redundant region $0 \le i_1 \le ... \le i_{k-1} < p$, and its symmetric extensions in the region, $[-p+1, p-1] \times [-p+1, p-1] \times ... \times [-p+1, p-1]$, which is a hypercube of side $(2p-1)$, centered at the origin, in the (k-1)-D space of the k-th order cumulant of $Y(n)$.

The steady-state solution in (80) does not assume that the SSM is in any particular form; in particular, it does not assume that the realization is minimal or even observable. However, this solution involves the inversion of the matrix $(I_k - \Phi_k)$. Explicit matrix inversion may be avoided either by using an iterative procedure or recasting the equation as a discrete Lyapunov matrix equation; for details, see Theorem 7 and the discussion following it; also, see Section 7 for an efficient algorithm to compute Kronecker products.

6.2 Time-Recursive Equations

From Theorem 5, (see also Theorem 7), the time-recursion for the zero-lag state-cumulant is given by

$$\mathbf{C}_{kx}(n+1) \;=\; \Phi_k \mathbf{C}_{kx}(n) + B_k \Gamma_{kw}(n) \tag{81}$$

If the measurement noise, $v(n)$, is Gaussian, then $\mathbf{C}_{kV}(n) = 0$, for $k > 2$. In standard second-order developments, we usually assume that the covariance matrix of the initial state $x(0)$ is known. In our case, we will assume that the k-th order cumulant vector of the initial state is known. Then, the state-cumulant can be propagated using (81), and the output cumulant can be computed using (79). Note that (81) and (79) constitute the state and output equations of a SSM, where the noise processes are replaced by statistical quantities. If the eigenvalues of Φ are λ_i, then, we know that the eigenvalues of Φ_k are given by $\lambda_{i_1}\lambda_{i_2}...\lambda_{i_k}$. Thus, stability of the model in (1) assures us of the stability of the propagation model in (81) and, hence, asymptotic indifference to the initial-state assumptions. Time-recursive computation of the state-cumulant vector using (81) and the output cumulant using (79) requires computation of the observability matrix and the first p impulse response coefficients of the original SSM in (1) and (2). Equation (79) does not provide a time-recursive equation for the output cumulants.

Replacing n by $n+1$ in (79), we have

$$\mathbf{C}_{kY}(n+1) = O_k \mathbf{C}_{kx}(n+1) + T_k \mathbf{C}_{kW}(n+1) + \mathbf{C}_{kV}(n+1) \tag{82}$$

If the SSM has time-varying matrices, then, (81) and (82) hold with time-varying O_k, T_k, Φ_k and B_k ; however, O_k is no longer an observability matrix [see (77) and (78)]. Using (81) in (82), yields,

$$\mathbf{C}_{kY}(n+1) = O_k \Phi_k \mathbf{C}_{kx}(n) + O_k B_k \Gamma_{kw}(n) + T_k \mathbf{C}_{kW}(n+1) + \mathbf{C}_{kV}(n+1) \tag{83}$$

Comparing (79) with (83), it is evident, that for an arbitrary (Φ, B, H) triple, we cannot derive time-recursive equations for $\mathbf{C}_{kY}(n+1)$ (because O and Φ need not be commutative). We will, therefore, assume that the SSM is in Kailath's *observability* form, [14, pp. 93-94], i.e., $O = I$ and hence $O_k = I_k$. This, of course, tacitly assumes that the SSM is observable. In this case, (with $O = I$), substituting for $\mathbf{C}_{kx}(n)$ from (79) into (83) yields

$$\mathbf{C}_{kY}(n+1) = \Phi_k[\mathbf{C}_{kY}(n) - T_k \mathbf{C}_{kW}(n) - \mathbf{C}_{kV}(n)] + B_k \Gamma_{kw}(n) + T_k \mathbf{C}_{kW}(n+1) + \mathbf{C}_{kV}(n+1) \tag{84}$$

which, on simplification, leads to:

Theorem 11. If the SSM in (1) and (2) is in observable form, then, in the time-invariant case, the output cumulant vector, $\mathbf{C}_{kY}$, can be computed recursively in time, as

$$\mathbf{C}_{kY}(n+1) = \Phi_k \mathbf{C}_{kY}(n) + \Theta_k(n) \tag{85}$$

where

$$\Theta_k(n) = T_k \mathbf{C}_{kW}(n+1) - \Phi_k T_k \mathbf{C}_{kW}(n) + \mathbf{C}_{kV}(n+1) - \Phi_k \mathbf{C}_{kV}(n) + B_k \Gamma_{kw}(n) \qquad (86)$$

The term $\Theta_k(n)$ in (85) and (86) depends only on the k-th order cumulant statistics of the input and measurement noises, and on the system impulse response; thus, (85) describes a time-recursive method for computing the cumulants of the output process $y(n)$. If the noise processes are stationary, then $\Theta_k(n)$ is independent of n, i.e.,

$$\Theta_k = (I_k - \Phi_k)(T_k \mathbf{C}_{kW} + \mathbf{C}_{kV}) + B_k \Gamma_{kw} \, .$$

Then, (85) resembles the state equation for a model excited by a constant non-random input vector (step function excitation). The stability of this system, in either case, is guaranteed by the assumed stability of the original model and the finiteness of the k-th order cumulant of the input.

The recursive solution in (85) depends crucially on the assumption that the SSM in (1)-(2) is observable and has been transformed to the observability form. Thus, given an arbitrary state-space triple, we must first find the similarity transformation (the observability matrix) that converts the triple into the observability form. When, the state-space model is, in fact, specified via an ARMA difference equation, then, the observable form is obtained very easily. In the MIMO case, if

$$\sum_{k=0}^{p} A(k) y(n-k) = \sum_{k=1}^{p} B(k) w(n-k) \qquad (87)$$

then, the matrices for the observable form of the SSM are given by

$$\Phi = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & & 0 \\ 0 & 0 & 0 & \cdots & I \\ -A(p) & -A(p-1) & -A(p-2) & \cdots & -A(1) \end{bmatrix}; \quad B = \begin{bmatrix} H(1) \\ H(2) \\ \vdots \\ H(p-1) \\ H(p) \end{bmatrix}; \quad \Psi^t = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \qquad (88)$$

where the $H(k)$'s are the impulse response matrices. If the sum on the right-hand side of (87) extends from 0 to p, as is usually assumed in a time-series model, the observation equation (2) has the additional term $H(0)w(n)$ on the right-hand side. In this case, the expression for the k-th order output cumulant, $C_{ky}(\tau_1, ..., \tau_{k-1})$ in (66), involves the additional term $[H(0) \otimes H(\tau_1) \otimes \cdots \otimes H(\tau_{k-1})]\Gamma_{kw}$.

In the MIMO case, the $A(k)$'s and the $B(k)$'s in (87) are respectively $n_y \times n_y$ and $n_y \times n_w$ matrices and the canonical representation given in (88) holds (see pp 685-90 and 791-806 in [21] for details); the elements in Φ , B and Ψ are respectively, $n_y \times n_y$, $n_y \times n_w$ and $n_y \times n_y$. The realization in (88) is observable, but not minimal [1]. Note that the MIMO model is stationary (stable) if the roots of $\det(\alpha(z))$ lie within the unit circle [11, p. 326], where $\alpha(z) = \sum_{k=0}^{p} A(k) z^{-k}$ is the AR polynomial matrix (the $a(k)$'s are $n_y \times n_y$

matrices). The model is invertible ('minimum-phase') if the roots of $\det(\beta(z))$ lie within the unit circle [11], where $\beta(z) = \sum_{k=0}^{q} B(k)z^{-k}$ is the MA polynomial matrix.

If the SSM matrices are time-varying, then the Kronecker state-cumulant vectors can be computed using the lag- and time-recursive methods developed in Section 5 (see, Fig. 2); the output cumulants can then be computed via Theorem 8. In the stationary time-invariant case, all the output cumulants in the non-redundant region (see Fig. 3a) can be computed in one shot via Theorem 10; cumulants at lags outside this region may be computed via the recursion equation, [c.f. (75) for k=4] and the symmetry properties of cumulants. When the SSM matrices are time-invariant, and the SSM is observable, but the noise processes are non-stationary, then Theorem 11 may be used to compute the output cumulants in the non-redundant region.

6.3 Applications

Parameter Estimation: Cumulant-Matching. One cumulant-based solution to the SISO ARMA system identification problem consists of matching the output cumulants of a proposed model to those of the observed output. In [26] (see also [15]) the spectrally-equivalent minimum-phase (SEMP) model is first obtained via correlation-based techniques. Since each of the p minimum-phase zeros may be reflected to its reciprocal location outside the unit circle, there are 2^p competing models. Then, the theoretical cumulant values are obtained using the Brillinger-Rosenblatt result, (16), i.e.,

$$c_{3T}(m,n) = \sum_{i=0}^{\infty} h(i)h(i+m)h(i+n) \tag{89}$$

where, in practice, the upper limit on the summation is replaced by $N \gg 0$, to yield an approximate value (justification: exponential stability implies decaying impulse response). Finally, the squared error between the sample cumulants of the observed output,

$$\bar{c}_{3y}(m,n) = \frac{1}{N}\sum_{i=1}^{N} y(i)y(i+m)y(i+n) \tag{90}$$

and the theoretical cumulant values given by (89),

$$\epsilon_T = \sum_m \sum_n [c_{3T}(m,n) - \bar{c}_{3y}(m,n)]^2 \, , \tag{91}$$

is computed for each of the 2^p models; the model which minimizes ϵ_T is then declared to be the true model. For this algorithm to work, the theoretical cumulants given by (89) have to be computed for a large range of the lags, m and n.

This procedure requires the explicit computation of the impulse response for each of the 2^p proposed models, followed by evaluation of the summation in (89) for each desired lag (m,n). With the model in the observable form, our steady-state solution in (80) is computationally cheap. The matrix Φ is fixed, since the AR parameters are assumed available; thus, the matrix inversion (for which we have indicated several efficient methods) needs to be done only once. Further, the vector B is easily obtained for each of the 2^p proposed models, (compute the first p values of the impulse response). Thus, the theoretical cumulant values for each of the proposed models are easily obtained, without any approximations.

Parameter Estimation: Closed-form solution. When the SSM is observable and in observable form, the model parameters may be estimated from the output cumulants.

Theorem 12. When the SISO SSM in (1) and (2) is time-invariant and in observable form, the plant matrices Φ and b can be estimated from the output cumulants.
Proof. Consider the k-th order output cumulant vector, $\mathbf{C}_{ky}(\tau_1, ..., \tau_{k-1})$. Let $\tau_1, ..., \tau_{k-2}$ be fixed and let $\tau_{k-1} = \tau$; for convenience, we will denote the corresponding output (scalar) cumulant by $C(..., \tau)$, where the ellipses denote the fixed lags. From Eq. (71) in Theorem 9, we know that the output cumulant $C(..., \tau) = \Psi\Phi^{\tau-1}\bar{G}$ is the impulse response of the SSM triple, $\{\Phi, \bar{G}, \psi\}$. Let M_1 and M_2 denote $p \times p$ Hankel matrices, with (i, j) elements $C(..., i+j-1)$ and $C(..., i+j)$, $i, j = 1, ..., p$. Then, we have [14, Sec. 2.2-2.3],

$$M_1 = \mathcal{O}(\psi, \Phi)\mathcal{C}(\Phi, \bar{G}) \tag{92}$$
$$M_2 = \mathcal{O}(\psi, \Phi)\Phi\mathcal{C}(\Phi, \bar{G}) \tag{93}$$

If the triple $\{\Phi, \bar{G}, \psi\}$ is minimal, and has the same order as the original SSM triple, $\{\Phi, b, \psi\}$ (which can always be arranged by a suitable choice of the fixed lags, $\tau_1, ..., \tau_{k-2}$, see [8], i.e., we assume that this slice of the output cumulant slice is a 'full-rank' slice), then, hence, M_1 is non-singular, and the Φ matrix may be obtained from

$$\Phi = M_2 M_1^{-1} = \mathcal{O}(\psi, \Phi)\Phi\mathcal{O}^{-1}(\psi, \Phi) \tag{94}$$

Note that this yields the matrix Φ corresponding to the observable realization.

The vector b, corresponding to the observable form, can also be obtained from the output cumulants via

$$b = \alpha S_{km}[I_k - \Phi_k]\mathbf{C}_{kY} \tag{95}$$

where the $(p-1)$ by p^k selector matrix S_{km} has a $p-1$ by $p-1$ identity matrix in columns $mp^{k-1} - p + 1$ through $mp^{k-1} - 1$ and zeros elsewhere; and $m = p - q + 1 \geq 1$. The scalar α is a scale factor which ensures that $h(1) = 1$. The derivation of (95) is given in Appendix B. □

Since we assumed that the SSM is in observable form, the vector b in (95) contains the first $q-1$ samples of the impulse response. If the underlying process is, in fact, an ARMA process given by (87), then, the AR parameters may be directly read off from the Φ matrix obtained in (94) [see (88)]. Once the IR coefficients have been obtained, the MA parameters may be obtained from

$$b(n) = \sum_{k=0}^{n-1} a(k)h(n-k), \quad n = 1, ..., q \tag{96}$$

Equation (95) for the special case $p = q$ with $m = 1$, and using third-order cumulants, was first given in [9], where Φ is obtained by applying Kung's balanced realization algorithm to M_1.

It may be shown, that the solution given by (95) is a state-space version of the 'q-slice' solution given in [25]. Theorem 12 requires knowledge of the MA order q; if the order is not known, we could assume $p = q$, and use some criterion to determine whether the estimated coefficients are zero (in which case, one would repeat the procedure with $q = p - 1$ and so on).

We expect that (95) can be extended to MIMO models as well. Estimation of the parameters of a multichannel MA model is discussed in [13], where a non-obvious extension of the SISO results in [6] is derived.

7. Computational Aspects

In this section, we will let F_k denote a general $p \times p$ matrix, rather than $F_k = F \otimes F_{k-1}$, as elsewhere in this report.

In Theorems 2-12 we have equations of the general form

$$\mathbf{C}_n = [F_1 \otimes F_2 \otimes \cdots \otimes F_k]\mathbf{C}_o = G\mathbf{C}_o \tag{97}$$

where the F_i's are $p \times p$ matrices, the $\mathbf{C}$'s are $p^k \times 1$ vectors and the subscripts o and n are mnemonics for 'old' and 'new' respectively. In the time-varying case, it does not make sense to store the time-varying matrix G, since the matrix G will vary both with the time and the lag indices. A simple-minded way of evaluating (97) would be to compute the matrix G as a Kronecker product of k matrices, and then compute the usual matrix product $G\mathbf{C}_o$. From the definition of Kronecker products, we see that computation of the Kronecker product of k $(p \times p)$ matrices requires $(k-1)p^{2k}$ flops. Computation of the matrix product $G\mathbf{C}_o$ requires p^{2k} flops. Thus, computation of $\mathbf{C}_n$ from $\mathbf{C}_o$ via (97) requires a total of kp^{2k} flops, which even for moderate values of p could be quite large (e.g., $p = 10$ and $k = 4$, requires 4×10^8 flops).

The above approach, however, does not take into account the special properties of Kronecker products. In the following, we develop a computational scheme that requires kp^{k+1} flops. For the example above (with $p = 10$ and $k = 4$), the required computation via the new scheme is only 10^{-3}, i.e., 0.1 % of that required by the simple-minded scheme.

Theorem 13. Let A be $m \times n$ and B be $p \times q$. Further, let Y be $mp \times 1$ and X be $nq \times 1$. Then,

$$Y = (A \otimes B)\, X \tag{98}$$

can be evaluated in $np(m + q)$ flops, with np units of intermediate storage.

Note that direct evaluation of (98) requires $2mnpq$ flops ($mnpq$ to evaluate $D = (A \otimes B)$, plus $mnpq$ flops to evaluate $Y = DX$), and $mnpq$ elements of storage for the intermediate variable (D).

Proof. From the definition of Kronecker products, we may write (98) as

$$Y = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_m \end{bmatrix} \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_n \end{bmatrix}; \tag{99}$$

where $\mathbf{X}_i$, $i = 1, ..., n$, are $q \times 1$ vectors, and $\mathbf{Y}_i, i = 1, ..., m$, are $p \times 1$ vectors. Thus, we have

$$\mathbf{Y}_i = \sum_{j=1}^{n} a_{ij}\mathbf{Z}_j, \quad i = 1, ..., m \tag{100}$$

where

$$\mathbf{Z}_j = B\mathbf{X}_j, \quad j = 1, ..., n \tag{101}$$

are $p \times 1$ vectors. Evaluation of each of the products $B\mathbf{X}_j$ requires pq flops; hence, evaluation of all the intermediate quantities $\mathbf{Z}_j$, $j = 1, ..., n$, requires npq flops, and np elements of storage. Once the $\mathbf{Z}_j$'s have been evaluated, evaluation of each $\mathbf{Y}_i$ via (100) requires an additional pn flops; and hence, mnp flops to evaluate all of the $\mathbf{Y}_i$'s, $i = 1, ..., m$. Thus, evaluation of the expression in (98) may be done in $np(m+q)$ flops, with np elements of intermediate storage. □

Corollary 3. If A_i, $i = 1, ..., k$, are $r \times r$ matrices, and X and Y are r^k vectors, then,

$$Y = [\otimes_{i=1}^{k} A_i]\ X \tag{102}$$

can be evaluated in kr^{k+1} flops, with r^k elements of storage for the intermediate variables.

Proof. For $k = 2$, the result follows immediately from Theorem 13, with $m = n = p = q = r$.

For $k > 2$, let $A = A_1$ and $B = \otimes_{i=2}^{k} A_i$ in Theorem 13; hence, $m = n = r$ and $p = q = r^{k-1}$. Then, in Theorem 13, once the $\mathbf{Z}_j$'s are available, Y can be computed in $r^2 r^{k-1} = r^{k+1}$ flops. Now

$$\mathbf{Z}_j = B\mathbf{X}_j = A_2 \otimes (A_3 \otimes \cdots \otimes A_k)\mathbf{X}_j$$

Hence, we can appeal to Theorem 13 again, with $m = n = r$ and $p = q = r^{k-2}$ to evaluate the $\mathbf{Z}_j$'s. Since there are r terms to be evaluated, we require $r.r^2.r^{k-2} = r^{k+1}$ flops. We continue the procedure until the kth step, where we will have to evaluate r^{k-1} products of the form Ax, where A is $r \times r$ and x is $r \times 1$; this step also requires r^{k+1} flops. Thus, (102) may be evaluated in a total of kr^{k+1} flops.

Storage is required for the intermediate quantities, the $\mathbf{Z}_j$'s. From (100), we note that the required storage is the dimension of the final output vector; hence, the required storage is r^k units. □.

We illustrate the procedure for $k = 3$ and $k = 4$: To evaluate $\mathbf{Y} = [A \otimes B \otimes C]\mathbf{X}$:

1. Let $\mathbf{X} = [\mathbf{X}_1^T, ..., \mathbf{X}_p^T]^T$. Further, let $\mathbf{X}_i = [\mathbf{X}_{i1}^T, ..., \mathbf{X}_{ip}^T]^T$. Note that $\mathbf{X}$, $\mathbf{X}_i$, $i = 1, .., p$, and $\mathbf{X}_{ij}$, $i, j = 1, ..., p$, are respectively p^3-, p^2- and p-element vectors. Let $\mathbf{Y}$ and $\mathbf{Z}$ be similarly partitioned.

2. $\mathbf{X}_{ij} = C\mathbf{X}_{ij}$, $i, j = 1, ..., p$.

3. $\mathbf{Z}_{ij} = \sum_{k=1}^{p} b_{jk}\mathbf{X}_{ik}$, $i, j = 1, ..., p$.

4. $\mathbf{Y}_i = \sum_{j=1}^{p} a_{ij}\mathbf{Z}_j$, $i = 1, ..., p$.

Each of the last three steps requires p^4 flops.

To evaluate $\mathbf{Y} = [A \otimes B \otimes C \otimes D]\mathbf{X}$:

1. Compute $\mathbf{Z}_j = [B \otimes C \otimes D]\mathbf{X}_j$, $j = 1, ..., p$ using the $k = 3$ algorithm.

2. Evaluate $\mathbf{Y}_i = \sum_{j=1}^{p} a_{ij}\mathbf{Z}_j$, $i = 1,..,p$.

The first step requires $3p^4 \times p$ flops, and the second step requires p^5 flops.

The moral of the story is that to efficiently evaluate equations, such as (97), which involve Kronecker products, one must exploit the special properties of Kronecker products, rather than use straight-forward matrix algebra. The direct evaluation of (97) using matrix products, requires kp^{2k} flops and p^{2k} units of storage. The algorithm presented above, requires only kp^{k+1} flops and p^k units of storage. Matrix representation methods, [19], [22], require less computation and storage than the direct method, but substantially more than the method presented above, because they cannot fully exploit the properties of Kronecker products. In the stationary, time-invariant case, symmetry properties of the cumulants may be exploited to further reduce the required computation and storage.

8. Simulations

In Sections 3-6, we derived time- and lag-recursive equations for the cumulants of the state and output processes of a SSM model. In this section, we use some of the recursive equations to compute output cumulants.

Figure 4a shows the impulse response of a tenth-order model, taken from [18]; the model parameters were obtained in [18] so as to fit the measured impulse response of a Bolt standard test shot of air-gun model 600B-20 inch^3, used in reflection seismology.

The SSM matrices given in [18] were converted to the observable form via MATLAB. Then, assuming that the input excitation is i.i.d., with third-order cumulant, $\gamma_{3w}(n) = 1$, eqs. (85) and (86) were used to recursively compute the output third-order cumulants in the non-redundant region (see Fig. 3b), with $\mathbf{C}_{3Y}(0) = \mathbf{0}$. Figure 4b shows the third-order output cumulant $C_{3y}(n; i, j)$, $-9 \leq i, j \leq 9$ at $n = 200$ (by which time the cumulants are in steady-state). We repeated the above procedure to compute fourth-order cumulants, with $\gamma_{4w} = 1$, and $\mathbf{C}_{4Y}(0) = \mathbf{0}$. 2-D slices of the fourth-order cumulant, $C_{4y}(n; i, j, k)$ with k fixed and $-9 \leq i, j \leq 9$ are shown in Fig. 5, at $n = 200$.

Figure 6 shows the diagonal slice of the third-order output cumulant, $C_{3y}(n; i, i)$, $n = 20, 21, ..., 200$, and $-9 \leq i \leq 9$. Here we assumed that the input process is non-stationary, with $\gamma_{3w}(n) = 0.9965^n$ (such a model for the input is typical of reflectivity sequences in reflection seismology, where the exponential decay roughly models absorption effects). An exponential taper, $s(n) = 1.0965^n$, has been applied to the output cumulants shown in the figure. Observe that, because $\gamma_{3w}(n)$ approaches zero as n increases, $C_{3y}(n; i, i)$ also approaches zero as n increases.

9. Conclusions

Kronecker products were used to obtain a compact and *uniform* representation for the cumulants of vector processes. This representation leads to elegant and *easily generalizable* expressions for the state and output cumulant vectors of state-space models. Further, the Kronecker product approach is computationally efficient compared to existing matrix representation methods. Time- and lag-recursive equations for the cumulants of the state and output processes of a MIMO SSM were derived using the Kronecker product approach. Closed-form expressions relating the output cumulant to the SSM parameters were also derived. Applications of these procedures to the SISO parameter estimation problem were discussed.

Since control systems operate in real time, we conjecture that the recursive equations developed in this report will be useful in developing state estimators based on higher-order statistics, and even optimal controllers based on higher-order statistics.

Acknowledgements

The work described in this report was performed at the University of Southern California, Los Angeles, under National Science Foundation Grant ECS-8602531. The authors wish to thank G. Giannakis and W. Wang for stimulating discussions on this and related topics.

References

[1] H. Akaike, "Stochastic Theory of Minimal Realizations", *IEEE Trans. on Auto. Contr.*, AC-19, 6, 667-73, 1974.

[2] J.W. Brewer, "Kronecker Products and Matrix Calculus in System Theory", *IEEE Trans. on Cir. Sys*, CAS-25, 772-781, 1978.

[3] D.R. Brillinger, "An Introduction to Polyspectra", *Ann. Math. Stat.*, pp. 1351-74, 1965.

[4] D.R. Brillinger and M. Rosenblatt, "Computation and Interpretation of k-th order spectra", in *Spectral Analysis of Time Series*, B. Harris, Ed., John Wiley and Sons, 189-232, 1967.

[5] G.B. Giannakis, *Signal Processing Using Higher-Order Statistics*, Ph.D. Thesis, U.S.C., 1987.

[6] G.B. Giannakis, "Cumulants: a Powerful Tool in Signal Processing", *Proc. IEEE*, Aug. 1987, pp 1333-4

[7] G.B. Giannakis, "A Kronecker Product Formulation of the Cumulant Based Realization of Stochastic Systems", *Proc. Amer. Control conf.*, Atlanta, GA, pp. 2096-2101, June 1988.

[8] G.B. Giannakis and J.M. Mendel, "Identification of Non-Minimum Phase Systems Using Higher-Order Statistics", *accepted for publication in IEEE Trans. on ASSP*, 1988.

[9] G.B. Giannakis and A. Swami, "New Results on State-Space and Input-Output Identification of Non-Gaussian Processes Using Cumulants", *Proc. SPIE Conf.*, San Diego, CA, Aug. 1987.

[10] G.H. Golub and Van Loan, *Matrix Computations*, John Hopkins University Press, Baltimore, MD, 1985.

[11] E.J. Hannan, *Multiple Time Series*, Wiley, New York, 1970.

[12] E. J. Hannan, "The Identification and Parametrization of ARMAX and State Space Forms", *Ecnometrica*, 44, 713-23, 1976.

[13] Y. Inouye, G. Giannakis and J.M. Mendel, "Parameter Estimation of Multichannel Moving Average Processes", *Proc. ICASSP-88*, 1252-55, New York, NY, 1988.

[14] T. Kailath, *Linear Systems*, Prentice-Hall, New Jersey, 1980.

[15] K.S. Lii and M. Rosenblatt, "Deconvolution and Estimation of Transfer Function Phase and Coefficients for Non-Gaussian Linear Processes", *Ann. Statist.*, v.10, 1195-1208, 1982.

[16] J.M. Mendel, "Lessons in Digital Estimation Theory", *Prentice-Hall*, New Jersey, 1987.

[17] J.M. Mendel, "Use of Higher-Order Statistics in Signal Processing and System Theory: A Short Perspective", *Proc. 8th Intl. Conf. on Math. Theory of Networks and Systems*, Phoenix, AZ, June 1987.

[18] J.M. Mendel, J. Kormylo, F. Aminzadeh, J.S. Lee and F. Habibi-Ashrafi, "A Novel Approach to Seismic Signal Processing and Modeling", *Geophysics*, v46, 1398-1414, 1981.

[19] J.M. Mendel and W. Wang, "Recursive Methods for Computation of One-Dimensional Cumulants", *Proc. American Control Conf.*, 2102-7, Atlanta, GA, June 1988.

[20] C.L. Nikias and M.R. Raghuveer, "Bispectrum Estimation: A Digital Signal Processing Framework", *Proc. IEEE*, v 75, pp 869-891, 1987.

[21] M.B. Priestly, *Spectral Analysis and Time Series*, Academic Press, London, 1981.

[22] M. Raghuveer, "Multichannel Bispectrum Estimation", *Proc. III IEEE-ASSP Workshop on Spectral Estimation and Modeling*, Boston, 1986, 21-24.

[23] Y. Rosen and B. Porat, "The Computation of the Fourth-Order Cumulants of an ARMA Process", submitted to *IEEE Trans. on ASSP*, April 1987.

[24] M. Rosenblatt, *Stationary Sequences and Random Processes*, Birkhaüser, Boston, 1985.

[25] A. Swami and J. Mendel, "Estimation of the Parameters of an ARMA System Using Only Output Cumulants", presented at the *Fourth IEEE ASSP Workshop on Spectrum Estimation and Modeling*, Minneapolis, MN, Aug. 3-5, 1988.

[26] J.K. Tugnait, "Identification of Nonminimum Phase Linear Stochastic Systems", *Proc. 23rd IEEE Conf. Decision and Control*, pp. 342-347, Las Vegas, 1984; also in *Automatica*, v.22 457-464, 1986.

Appendix A: Kronecker Products

The Kronecker product of a $(p \times q)$ matrix $A = \{a_{ij}\}$ and an $(m \times n)$ matrix $B = \{b_{ij}\}$ is the $(pm \times qn)$ matrix, $\{a_{ij}B\}$, denoted by $A \otimes B$. The $[(i_1-1)m+i_2, (j_1-1)n+j_2]$ element of $A \otimes B$ is $a(i_1, j_1)b(i_2, j_2)$, for $i_1 = 1, ..., p$, $j_1 = 1, ..., q$, $i_2 = 1, ..., m$ and $j_2 = 1, ..., n$. A review of Kronecker product theory may be found in [2].

We will find the **vec** operator [2] also useful; if A is $m \times n$, then, $\mathbf{vec}(A)$ is an $mn \times 1$ vector obtained from A by lexicographic ordering, i.e., column-wise stacking. Thus, if $A = [c_1, ..., c_n]$, then,

$$\mathbf{vec}(A) = [c_1', ..., c_n']'$$

We *define* an $\mathbf{unvec}_{m,n}$ operator which converts an mn-element vector into an $m \times n$ matrix, such that

$$\mathbf{unvec}_{m,n}[\mathbf{x}] = X_{m \times n} \Longrightarrow \mathbf{vec}\ [X_{m \times n}] = \mathbf{x} \tag{103}$$

We will let $[\otimes_{i=1}^{n} A(i)] = A(1) \otimes A(2) \otimes \cdots \otimes A(k)$.

We will need the following results from [2] (compatible matrix dimensions are assumed):

[P1] $(A + B) \otimes (C + D) = A \otimes C + A \otimes D + B \otimes C + B \otimes D$; hence,

$$[\sum_{i=1}^{m} A(i)] \otimes [\sum_{j=1}^{n} B(j)] = \sum_{i=1}^{m} \sum_{j=1}^{n} [A(i) \otimes B(j)] \ ,$$

[P2] $(A \otimes B)(C \otimes D) = AC \otimes BD$; hence,

$$\prod_{i=1}^{n} [A(i) \otimes B(i)] = [\prod_{i=1}^{n} A(i)] \otimes [\prod_{i=1}^{n} B(i)] \ ,$$

and

$$[\otimes_{i=1}^{n} A(i)][\otimes_{i=1}^{n} B(i)] = \otimes_{i=1}^{n} A(i)B(i) \ .$$

[P3] $(A \otimes B) \otimes (C \otimes D) = A \otimes B \otimes C \otimes D$.

[P4] If α_i is an eigenvector of A associated with eigenvalue λ_i, and β_j is an eigenvector of B associated with eigenvalue μ_j, then, $\alpha_i \otimes \beta_j$ is an eigenvector of $A \otimes B$, associated with eigenvalue $\lambda_i \mu_j$.

[P5] $(A \otimes B)^{\dagger} = A^{\dagger} \otimes B^{\dagger}$ where $A^{\dagger}$ denotes the pseudo-inverse of A, and $(A \otimes B)^t = A^t \otimes B^t$.

[P6] $\mathbf{vec}(PAQ) = (Q^t \otimes P)\mathbf{vec}(A)$.

[P7] If A is $p \times q$ and B is $m \times n$, then,

$$B \otimes A = U_{m \times p}(A \otimes B) U_{q \times n}$$

where the U matrices are permutation matrices. In particular, if A and B are column vectors, (i.e., $q = n = 1$), then, $B \otimes A = U_{m \times p}(A \otimes B)$. The $(mp \times mp)$ permutation matrix $U_{m \times p}$ is given by [2, eq. (4)]

$$U_{m \times p} = \sum_{i=1}^{m} \sum_{k=1}^{p} E_{ik}^{m \times p} \otimes E_{ki}^{p \times m} \tag{104}$$

where $E_{ik}^{m \times p}$ is the $m \times p$ elementary matrix with unity in element (i, k) and zeros elsewhere. Matrix $U_{m \times p}$ has unity values in elements $[(i-1)p + k, (k-1)m + i]$, $i = 1, ..., m$ and $k = 1, ..., p$. Furthermore,

$$U_{m \times p} = U_{p \times m}^{T} = U_{p \times m}^{-1} \,.$$

We will let $A_{k+1} = A \otimes A_k = A_k \otimes A$, with $A_1 = A$; if A is a square matrix, then, we will also let $A_0 = I$. This differs from the $A^{[k]}$ notation in [2].

Appendix B: Proofs

Notation. If $\mathbf{C}$ is a p^k element vector, we will let $\mathbf{C}[i_1, i_2, ..., i_k]$, $i_1, ..., i_k = 1, ..., p$, denote its $\big((i_1 - 1)p^{k-1} + (i_2 - 1)p^{k-2} + \cdots + i_k\big)$-th element. Essentially, the p^k-element vector is 'treated' as a k-dimensional array. Similarly, if M is a $p^k \times p^k$ matrix, we will represent its $\big((i_1 - 1)p^{k-1} + (i_2 - 1)p^{k-2} + \cdots + i_k, (j_1 - 1)p^{k-1} + (j_2 - 1)p^{k-2} + \cdots + j_k\big)$ element by $M([i_1, ..., i_k], [j_1, ..., j_k])$, $i_1, ..., i_k, j_1, ..., j_k = 1, ..., p$. The length p will usually be obvious from the context.

Theorem 1. We must verify that the elements of $\mathbf{C}_{ky}$, $k = 2, 3, 4$ defined in (20)-(22) are the cross-cumulants of the vector elements as defined in (17)-(19) and that all possible cross-cumulants are contained in them.

Concatenate (17) for $i = 1, ..., p$ and $j = 1, ..., p$, with j varying faster than i, and let $\mathbf{C}_{2y}(t; \tau)$ denote the resulting vector. Using the definition of Kronecker products leads to (20).

Similarly, concatenate (18) for $i = 1, ..., p, j = 1, ..., p, k = 1, ..., p$, with k varying fastest and i slowest. Let $\mathbf{C}_{3y}(t; t_1, t_2)$ denote the vector so obtained. Then, from the definition of Kronecker products, we obtain (21).

The fourth-order case is a little bit more involved. In particular, note that $E\{y(t) \otimes y(t + t_1) \otimes y(t + t_2) \otimes y(t + t_3)\}$ yields the vector of fourth-order moments, not cumulants.

For convenience, we will rewrite (22) as

$$M_0 = M_1 - M_2 - P_p^T M_3 - P_p M_4 \tag{105}$$

where the terms M_i, $i = 0, 1, 2, 3, 4$, are obvious. We must show that $M_0[i, j, k, l] = \text{cum}(y_i(t), y_j(t + t_1), y_k(t + t_2), y_l(t + t_3))$.

Using the definition of Kronecker products, we obtain

$$M_1[i, j, k, l] = E\{y_i(t) y_j(t + t_1) y_k(t + t_2) y_l(t + t_3)\} \tag{106}$$

and

$$M_2[i, j, k, l] = E\{y_i(t) y_j(t + t_1)\} E\{y_k(t + t_2) y_l(t + t_3)\} \tag{107}$$

From [P7], matrix $U_{p^2 \times p}$ has unity entries in elements $((j' - 1)p + l, (l - 1)p^2 + j')$, $j' = 1, ..., p^2$ and $l = 1, ..., p$. Let $j' = (j - 1)p + k$, $j, k = 1, ..., p$. Then, the $p^3 \times p^3$ matrix $U_{p^2 \times p}$ has unity entries in elements $([j, k, l], [l, j, k])$. Hence, using the definition of Kronecker products, permutation matrix $P_p = [I \otimes U_{p^2 \times p}]$ is defined by

$$P_p\left([i, j, k, l], [i, l, j, k]\right) = 1, \quad i, j, k, l = 1, ..., p \tag{108}$$

Now, if a permutation matrix P has unity entry in element (i, j), and, if $b = Pa$, then $b(i) = a(j)$. Hence, if a and b are p^4-element vectors, then,

$$b = P_p a \Longrightarrow b[i, j, k, l] = a[i, l, j, k] \tag{109}$$

which follows from (108). Furthermore, if $d = P_p^T a$, then, from (108) and (109), we have $d(i,l,j,k) = a(i,j,k,l)$ which may be re-written, via a permutation of the indices j,k,l, as

$$d = P_p^T a \Longrightarrow d[i,j,k,l] = a[i,k,l,j] \tag{110}$$

Since $M_3[i,k,l,j] = E\{y_i(t)y_k(t+t_2)\}E\{y_l(t+t_3)y_j(t+t_1)\}$, we obtain, using (110),

$$(P_p^T M_3)[i,j,k,l] = E\{y_i(t)y_k(t+t_2)\}E\{y_l(t+t_3)y_j(t+t_1)\} \tag{111}$$

Now, $M_4[i,l,j,k] = E\{y_i(t)y_l(t+t_3)\}E\{y_j(t+t_1)y_k(t+t_2)\}$. Therefore, from (109),

$$(P_p M_4)[i,j,k,l] = E\{y_i(t)y_l(t+t_3)\}E\{y_j(t+t_1)y_k(t+t_2)\} \tag{112}$$

From (105)-(107), (111) and (112), we can conclude that the $[i,j,k,l]$ element of the vector $\mathbf{C}_{4y}(t;t_1,t_2,t_3)$ is indeed the fourth-order cumulant of $\{y_i(t), y_j(t+t_1), y_k(t+t_2), y_l(t+t_3)\}$. Hence, equation (22) is a valid representation for the fourth-order cumulants of the vector process $y(t)$. □

CP1. Let $y_i = \Lambda_i x_i$, $i = 1,...,k$ and let $\mathbf{C} = \mathrm{cum}(y_1,...,y_k)$. Further, let $\lambda_{i;j,l}$ denote the (j,l) element of Λ_i. Then, from our definition in Section 4,

$$\begin{aligned}
\mathbf{C}[i_1,...,i_k] &= \mathrm{cum}(y_{1,i_1},...,y_{k,i_k}) \\
&= \mathrm{cum}\left(\sum_{j_1=1}^{p} \lambda_{1;i_1,j_1} x_{1,j_1},...,\sum_{j_k=1}^{p} \lambda_{k;i_k,j_k} x_{k,j_k}\right) \\
&= \sum_{j_1=1}^{p} \cdots \sum_{j_k=1}^{p} \lambda_{1;i_1,j_1} \cdots \lambda_{k;i_k,j_k} \mathrm{cum}(x_{1,j_1},...,x_{k,j_k})
\end{aligned} \tag{113}$$

where we have used [CP1] and [CP3] for *scalar random variables* [24, Ch II.2]. From the definition of Kronecker products, we have

$$\begin{aligned}
&(\Lambda_1 \otimes \cdots \otimes \Lambda_k)\mathrm{cum}(x_1,...,x_k)[i_1,...,i_k] \\
&= \sum_{j_1=1}^{p} \cdots \sum_{j_k=1}^{p} \lambda_{1;i_1,j_1} \cdots \lambda_{k;i_k,j_k} \mathrm{cum}(x_1,...,x_k)[j_1,...,j_k] \\
&= \sum_{j_1=1}^{p} \cdots \sum_{j_k=1}^{p} \lambda_{1;i_1,j_1} \cdots \lambda_{k;i_k,j_k} \mathrm{cum}(x_{1,j_1},...,x_{k,j_k})
\end{aligned} \tag{114}$$

where the last equality follows from our definition in Section 4. Equations (113) and (114) establish [CP1]. □

CP2. Let $\mathbf{C} = \mathrm{cum}(x_1,...,x_k)$. Then, by definition,

$$\mathbf{C}[i_1,...,i_k] = \mathrm{cum}(x_{1,i_1},...,x_{k,i_k}) \tag{115}$$

From property [P7], the permutation matrix $U_{p^{k-j}\times p^j}$ has unity entries in elements $((i-1)p^j+r,(r-1)p^{k-j}+i)$, $i=1,..,p^{k-j}$, $r=1,...,p$. Let $i=(i_1-1)p^{k-j-1}+\cdots+(i_{k-j}-1)+1$, with $i_1,...,i_{k-j}=1,...,p$ and $l=(l_1-1)p^{j-1}+\cdots+(l_j-1)+1$, with $l_1,...,l_j=1,...,p$. Then, the permutation matrix $U_{p^{k-j}\times p^j}$ has unity entries in elements

$$([i_1,...,i_{k-j},l_1,...,l_j],[l_1,...,l_j,i_1,...,i_{k-j}])$$

Hence,

$$\begin{aligned} U_{p^{k-j}\times p^j}&\mathbf{C}[i_1,...,i_{k-j},l_1,...,l_j] \\ &= \operatorname{cum}\left(x_{1,l_1},...,x_{j,l_j},x_{j+1,i_1},...,x_{k,i_{k-j}}\right) \\ &= \operatorname{cum}\left(x_{j+1,i_1},...,x_{k,i_{k-j}},x_{1,l_1},...,x_{j,l_j}\right) \end{aligned} \tag{116}$$

since the cumulants of scalar random variables are symmetric in their arguments. [CP2] follows from (115) and (116). □

Theorem 2. Let $\tau_0=0$; then,

$$\begin{aligned} &\mathbf{C}_{ky}(t;\tau_1,...,\tau_{k-1}) \\ &\quad= \operatorname{cum}(y(t),y(t+\tau_1),...,y(t+\tau_{k-1})) \\ &\quad= \operatorname{cum}\left(\sum_{u_0} H(t+\tau_0,u_0)w(u_0),...,\sum_{u_{k-1}} H(t+\tau_{k-1},u_{k-1})w(u_{k-1})\right) \\ &\quad= \sum_{u_0}\cdots\sum_{u_{k-1}} \operatorname{cum}(H(t+\tau_0,u_0)w(u_0),...,H(t+\tau_{k-1},u_{k-1})w(u_{k-1})) \\ &\quad= \sum_{u_0}\cdots\sum_{u_{k-1}} [H(t+\tau_0,u_0)\otimes\cdots\otimes H(t+\tau_{k-1},u_{k-1})]\operatorname{cum}(w(u_0),...,w(u_{k-1})) \\ &\quad= \sum_{u_0}\cdots\sum_{u_{k-1}} [H(t+\tau_0,u_0)\otimes\cdots\otimes H(t+\tau_{k-1},u_{k-1})]\Gamma_{kw}(u_0)\delta(u_0-u_1)\cdots\delta(u_0-u_{k-1}) \\ &\quad= \sum_{u}[H(t,u)\otimes H(t+\tau_1,u)\otimes\cdots\otimes H(t+\tau_{k-1},u)]\Gamma_{kw}(u) \end{aligned} \tag{117}$$

where we have used (39), [CP3], [CP1] and (40). In the time-invariant case, $H(n,u)=H(n-u)$, and (42) follows immediately from (117). □

Theorem 3. From (10), we have,

$$x(n)=\sum_{k=0}^{n-1} A(n,k+1)B(k)w(k)+A(n,0)x(0)$$

where, by assumption, the random vector $x(0)$ is independent of the $w(j)$'s. Hence, by [CP5], the cumulant of the process $x(n)$ is the sum of the cumulants, due separately to the input and the initial state. We will use the subscripts *zis* (zero initial state) and *zin* (zero input) to represent these two terms.

Under zero-input conditions, the state vector is given by $x(n) = A(n,0)x(0)$, $n \geq 0$. Hence, we have

$$
\begin{aligned}
&\mathbf{C}_{kx,zin}(n-m;0,...,0) \\
&\quad = \mathrm{cum}_{zin}(x(n-m),...,x(n-m)) \\
&\quad = \mathrm{cum}(A(n-m,0)x(0),....,A(n-m,0)x(0)) \\
&\quad = [\otimes_{l=0}^{k-1} A(n-m,0)]\mathbf{C}_{kx}(0;0,...,0)
\end{aligned}
\tag{118}
$$

where we have used [CP1].

Furthermore, with $i_0 = 0$, and $m = -\min(i_0, i_1, ..., i_{k-1})$, we have

$$
\begin{aligned}
&\mathbf{C}_{kx,zin}(n;i_1,...,i_{k-1}) \\
&\quad = \mathrm{cum}_{zin}(x(n+i_0),...,x(n+i_{k-1})) \\
&\quad = \mathrm{cum}(A(n,0)x(0),A(n+i_1,0)x(0),...,A(n+i_{k-1},0)x(0)) \\
&\quad = [\otimes_{l=0}^{k-1} A(n+i_l,0)]\mathbf{C}_{kx}(0;0,...,0) \\
&\quad = [\otimes_{l=0}^{k-1} A(n+i_l,n-m)A(n-m,0)]\mathbf{C}_{kx}(0;0,...,0) \\
&\quad = [\otimes_{l=0}^{k-1} A(n+i_l,n-m)][\otimes_{l=0}^{k-1} A(n-m,0)]\mathbf{C}_{kx}(0;0,...,0) \\
&\quad = [\otimes_{l=0}^{k-1} A(n+i_l,n-m)]\mathbf{C}_{kx,zin}(n-m;0,...,0)
\end{aligned}
\tag{119}
$$

where we have used [CP1], (9), [P2] and (118).

The contribution due to the input term is obtained from Theorem 2, (with, $H(n,k) = A(n,k+1)B(k)$, $k \leq n-1$, $m = -\min(0,i_1,...,i_{k-1})$ and $i_0 = 0$), as

$$
\begin{aligned}
&\mathbf{C}_{kx,zis}(n;i_1,...,i_{k-1}) \\
&\quad = \sum_j \left[\otimes_{l=0}^{k-1} A(n+i_l,j+1)B(j)\right]\Gamma_{kw}(j) \\
&\quad = \sum_j \left[\otimes_{l=0}^{k-1} A(n+i_l,j+1)\right] B_k(j)\Gamma_{kw}(j) && (120) \\
&\quad = \sum_j \left[\otimes_{l=0}^{k-1} A(n+i_l,n-m)A(n-m,j+1)\right] B_k(j)\Gamma_{kw}(j) \\
&\quad = \sum_j \left[\otimes_{l=0}^{k-1} A(n+i_l,n-m)\right]\left[\otimes_{l=0}^{k-1} A(n-m,j+1)\right] B_k(j)\Gamma_{kw}(j) \\
&\quad = \left[\otimes_{l=0}^{k-1} A(n+i_l,n-m)\right]\mathbf{C}_{kx,zis}(n-m;0,...,0) && (121)
\end{aligned}
$$

where we have used property [P2], (9) and (120) to obtain the last equality.

Finally, from (119) and (121), we obtain

$$
\mathbf{C}_{kx}(n;i_1,...,i_{k-1}) = \left[\otimes_{l=0}^{k-1} A(n+i_l,n-m)\right]\mathbf{C}_{kx}(n-m;0,...,0)
\tag{122}
$$

which establishes the theorem. □

Theorem 4. From (122), we have, with $i_0 = m_0 = 0$, and $0 \leq m_l \leq i_l$, $l = 1, ..., k-1$ (hence, $m = -\min(i_0, ..., i_{k-1}) = 0$),

$$
\begin{aligned}
&\mathbf{C}_{kx}(n; i_1, ..., i_{k-1}) \\
&\quad = \left[\otimes_{l=0}^{k-1} A(n+i_l, n)\right] \mathbf{C}_{kx}(n; 0, ..., 0) \\
&\quad = \left[\otimes_{l=0}^{k-1} A(n+i_l, n+i_l-m_l) A(n+i_l-m_l, n)\right] \mathbf{C}_{kx}(n; 0, ..., 0) \\
&\quad = \left[\otimes_{l=0}^{k-1} A(n+i_l, n+i_l-m_l)\right] \left[\otimes_{l=0}^{k-1} A(n+i_l-m_l, n)\right] \mathbf{C}_{kx}(n; 0, ..., 0) \\
&\quad = \left[\otimes_{l=0}^{k-1} A(n+i_l, n+i_l-m_l)\right] \mathbf{C}_{kx}(n; i_1-m_1, ..., i_{k-1}-m_{k-1})
\end{aligned}
$$

where we have used (9), property [P2] and (122). □

Theorem 5. First, we will establish the theorem for the zero-lag case. From the definition of cumulants of vector processes, we have

$$
\begin{aligned}
\mathbf{C}_{kx}(n+1; 0, ..., 0) &= \mathrm{cum}\,(x(n+1), ..., x(n+1)) \\
&= \mathrm{cum}\,(\Phi(n)x(n) + B(n)w(n), ..., \Phi(n)x(n) + B(n)w(n)) \\
&= \mathrm{cum}\,(\Phi(n)x(n), ..., \Phi(n)x(n)) + \mathrm{cum}\,(B(n)w(n), ..., B(n)w(n)) \\
&= \Phi_k(n)\mathrm{cum}\,(x(n), ..., x(n)) + B_k(n)\mathrm{cum}\,(w(n), ..., w(n)) \\
&= \Phi_k(n)\mathbf{C}_{kx}(n; 0, ..., 0) + B_k(n)\Gamma_{kw}(n) \qquad (123)
\end{aligned}
$$

where we have used (1), [CP5] and [CP1].

From (122), we have, with $i_0 = 0$, and $i_l \geq 0$, $l = 1, .., k-1$ (hence, $m = -\min(i_0, ..., i_{k-1}) = 0$),

$$
\begin{aligned}
&\mathbf{C}_{kx}(n+1; i_1, ..., i_{k-1}) \\
&\quad = \left[\otimes_{l=0}^{k-1} A(n+1+i_l, n+1)\right] \mathbf{C}_{kx}(n+1; 0, ..., 0) \\
&\quad = \left[\otimes_{l=0}^{k-1} A(n+1+i_l, n+1)\right] [\Phi_k(n)\mathbf{C}_{kx}(n; 0, ..., 0) + B_k(n)\Gamma_{kw}(n)] \\
&\quad = \left[\otimes_{l=0}^{k-1} A(n+1+i_l, n+1)\Phi(n)\right] \mathbf{C}_{kx}(n; 0, ..., 0) + D(i_1, ..., i_{k-1}) \\
&\quad = \left[\otimes_{l=0}^{k-1} A(n+1+i_l, n)\right] \mathbf{C}_{kx}(n; 0, ..., 0) + D(i_1, ..., i_{k-1}) \\
&\quad = \left[\otimes_{l=0}^{k-1} \Phi(n+i_l) A(n+i_l, n)\right] \mathbf{C}_{kx}(n; 0, ..., 0) + D(i_1, ..., i_{k-1}) \\
&\quad = \left[\otimes_{l=0}^{k-1} \Phi(n+i_l)\right] \left[\otimes_{l=0}^{k-1} A(n+i_l, n)\right] \mathbf{C}_{kx}(n; 0, ..., 0) + D(i_1, ..., i_{k-1}) \\
&\quad = \left[\otimes_{l=0}^{k-1} \Phi(n+i_l)\right] \mathbf{C}_{kx}(n; i_1, .., i_{k-1}) + D(i_1, ..., i_{k-1})
\end{aligned}
$$

where we have used (123), [P2], the definition of $D(i_1, ..., i_{k-1})$ in (47), (7), (8), and (122). □.

Corollary 1. Follows immediately from [CP2], with $x_1 = x(n)$, $x_2 = x(n+i_1), ..., x_k = x(n+i_{k-1})$, and our definition $\mathbf{C}_{kx}(n; i_1, ..., i_{k-1}) = \mathrm{cum}\,(x(n), x(n+i_1), ..., x(n+i_{k-1}))$.

Note that (52) holds for *all* lags. □

Theorem 6. With $i_0 = 0$ and $m = -i_l < -\min(i_0, ..., i_{l-1}, i_{l+1}, ..., i_{k-1})$, in (43) of Theorem 3, we obtain,

$$\begin{aligned}
&\mathbf{C}_{kx}(n; i_1, ..., i_{k-1}) \\
&\quad = [\otimes_{l=0}^{k-1} A(n+i_l, n-m)]\mathbf{C}_{kx}(n-m; 0, ..., 0) \\
&\quad = [\otimes_{l=0}^{j-1} \Phi(n+i_l-1)A(n-1+i_l, n-m)] \otimes IA(n+i_j, n-m) \otimes \\
&\quad\quad [\otimes_{l=j+1}^{k-1} \Phi(n+i_l-1)A(n-1+i_l, n-m)]\mathbf{C}_{kx}(n-m; 0, ..., 0) \\
&\quad = [\otimes_{l=0}^{j-1} \Phi(n+i_l-1)] \otimes I \otimes [\otimes_{l=j+1}^{k-1} \Phi(n+i_l-1)] \\
&\quad\quad [\otimes_{l=0}^{j-1} A(n+i_l-1, n-m)] \otimes A(n+i_j, n-m) \otimes [\otimes_{l=j+1}^{k-1} A(n+i_l-1, n-m)] \\
&\quad\quad \mathbf{C}_{kx}(n-m; 0, ..., 0)
\end{aligned}$$

where we have used (8) and [P2].
With $m' = -(i_l+1) \leq -\min(i_0, ..., i_{l-1}, i_{l+1}, ..., i_{k-1})$ (hence $m' = m-1$) in (122), we obtain,

$$\begin{aligned}
&\mathbf{C}_{kx}(n-1; i_1, ..., i_{j-1}, i_j+1, i_{j+1}, ..., i_{k-1}) \\
&\quad = [\otimes_{l=0}^{j-1} A(n-1+i_l, n-1-(m-1))] \otimes A(n-1+i_j+1, n-1-(m-1)) \\
&\quad\quad \otimes [\otimes_{l=j+1}^{k-1} A(n-1+i_l, n-1-(m-1))]\mathbf{C}_{kx}(n-1-(m-1); 0, ..., 0) .
\end{aligned}$$

Theorem 6 follows from the last two equations. □

Equation (55). From (43) of Theorem 3, we have for lags inside R_{k-1}, with $i_0 = 0$ and $m = -i_{k-1} < -\min(i_0, ..., i_{k-2})$,

$$\begin{aligned}
&\mathbf{C}_{kx}(n; i_1, ..., i_{k-1}) \\
&\quad = [\otimes_{l=0}^{k-1} A(n+i_l, n-m)]\mathbf{C}_{kx}(n-m; 0, ..., 0) \\
&\quad = [\otimes_{l=0}^{k-1} A(n+i_l, n-m)]\Phi_k^{-1}(n-m)[\mathbf{C}_{kx}(n-m+1; 0, ..., 0) - B_k(n-m)\Gamma_{kw}(n-m)] \\
&\quad = [\otimes_{l=0}^{k-1} A(n+i_l, n-m)\Phi^{-1}(n-m)][\mathbf{C}_{kx}(n-m+1; 0, ..., 0) - B_k(n-m)\Gamma_{kw}(n-m)] \\
&\quad = [\otimes_{l=0}^{k-1} A(n+i_l, n-m+1)][\mathbf{C}_{kx}(n-m+1; 0, ..., 0) - B_k(n-m)\Gamma_{kw}(n-m)] \\
&\quad = [I_{k-1} \otimes \Phi^{-1}(n+i_{k-1})][\otimes_{l=0}^{k-2} A(n+i_l, n-m+1)] \otimes A(n+i_{k-1}+1, n-m+1) \\
&\quad\quad [\mathbf{C}_{kx}(n-m+1; 0, ..., 0) - B_k(n-m)\Gamma_{kw}(n-m)] \\
&\quad = [I_{k-1} \otimes \Phi^{-1}(n+i_{k-1})]\mathbf{C}_{kx}(n; i_1, ..., i_{k-2}, i_{k-1}+1) \\
&\quad - [\otimes_{l=0}^{k-1} A(n+i_l, n+i_{k-1}+1)]B_k(n+i_{k-1})\Gamma_{kw}(n+i_{k-1})]
\end{aligned}$$

where we have used Theorem 3, (45) of Theorem 5, property [P2], (7), (8), $m = -i_{k-1}$ and (122), with $m' = -(i_{k-1}+1) = m-1 \leq -\min(i_0, ..., i_{k-2})$. □

Lemma 1. From the definition of the third-order cumulants of stationary random processes, we have

$$\mathbf{C}_{3x}(i,j) \quad = \quad E\{x(n) \otimes x(n+i) \otimes x(n+j)\}$$

$$
\begin{aligned}
&= E\{x(n) \otimes U_{p\times p}[x(n+j) \otimes x(n+i)]\} \\
&= [I \otimes U_{p\times p}]E\{x(n) \otimes x(n+j) \otimes x(n+i)\} \\
&= [I \otimes U_{p\times p}]\mathbf{C}_{3x}(j,i) \qquad (124)
\end{aligned}
$$

where we have used [P2] and [P7]. Similarly, we obtain

$$
\begin{aligned}
\mathbf{C}_{3x}(i,j) &= E\{x(n) \otimes x(n+i) \otimes x(n+j)\} \\
&= U_{p^2\times p}E\{x(n+j) \otimes x(n) \otimes x(n+i)\} \\
&= U_{p^2\times p}\mathbf{C}_{3x}(-j, i-j) \qquad (125) \\
&= U_{p^2\times p}[I \otimes U_{p\times p}]\mathbf{C}_{3x}(i-j, -j) \qquad (126)
\end{aligned}
$$

where the last equality follows by using (124). Further,

$$
\begin{aligned}
\mathbf{C}_{3x}(i,j) &= E\{x(n) \otimes x(n+i) \otimes x(n+j)\} \\
&= U_{p\times p^2}E\{x(n+i) \otimes x(n+j) \otimes x(n)\} \\
&= U^T_{p^2\times p}\mathbf{C}_{3x}(j-i, -i) \qquad (127) \\
&= U^T_{p^2\times p}[I \otimes U_{p\times p}]\mathbf{C}_{3x}(-i, j-i) \qquad (128)
\end{aligned}
$$

where we have used [P7] and (124). Thus, all the identities in the Lemma have been established. □

Theorem 9. Substituting for the Kronecker state-cumulant vector from (59) into (66), we obtain for $\tau_l \geq 0$, $l = 1, ..., k-1$,

$$
\begin{aligned}
\mathbf{C}_{ky}(\tau_1, ..., \tau_{k-1}) &= \Psi_k[I \otimes \Phi^{\tau_1} \otimes ... \otimes \Phi^{\tau_{k-2}} \otimes \Phi^{\tau_{k-1}}]\mathbf{C}_{kx}(0, ..., 0) \\
&= [\Psi \otimes \Psi\Phi^{\tau_1} \otimes ... \otimes \Psi\Phi^{\tau_{k-2}} \otimes \Psi\Phi^{\tau_{k-1}}]\mathbf{C}_{kx}(0, ..., 0) \\
&= [I \otimes I \otimes ... \otimes I \otimes \Psi\Phi^{\tau_{k-1}-1}] \\
&\quad \times[\Psi \otimes \Psi\Phi^{\tau_1} \otimes ... \otimes \Psi\Phi^{\tau_{k-2}} \otimes \Phi]\mathbf{C}_{kx}(0, ..., 0) \\
&= [I_{k-1} \otimes \Psi\Phi^{\tau_{k-1}-1}]G_{k;\tau_1,...,\tau_{k-2}} \qquad (129)
\end{aligned}
$$

where

$$
\begin{aligned}
G_{k;\tau_1,...,\tau_{k-2}} &= [\Psi \otimes \Psi\Phi^{\tau_1} \otimes ... \otimes \Psi\Phi^{\tau_{k-2}} \otimes \Phi]\mathbf{C}_{kx}(0, ..., 0) \\
&= [\Psi_{k-1} \otimes I][I \otimes \Phi^{\tau_1} \otimes .... \otimes \Phi^{\tau_{k-2}} \otimes \Phi]\mathbf{C}_{kx}(0, ..., 0) \\
&= [\Psi_{k-1} \otimes I]\mathbf{C}_{kx}(\tau_1, ..., \tau_{k-2}, 1) \qquad (130)
\end{aligned}
$$

where the last line follows from (59). □

Theorem 12. We will prove the theorem only for the third-order cumulant. Extension to arbitrary orders is straight-forward. Let S_{3m} denote the $(p-1 \times p^3)$ selector matrix which has a $(p-1 \times p-1)$ identity matrix in columns $mp^2 - p + 1$ through $mp^2 - 1$ and zeros elsewhere. The variable m will be fixed in the range $[1, p-1]$.

Define the $(p-1)$ element vector

$$\beta_{3m} = S_{3m}(I_3 - \Phi_3)\mathbf{C}_{3Y} \tag{131}$$

Also let $\alpha(k) = -a(p+1-k)$, $k = 1, ..., p$. We need to show that the elements of β_{3m} are proportional to the impulse response coefficients.

Using the definitions of Kronecker products, we obtain

$$\begin{aligned} & S_{3m}\Phi_3[i, mp^2 + (k-1)p + i + 1] \\ & \quad = \begin{cases} \alpha(k), & k = 1, ..., p; \quad i = 1, ..., p-1 \\ 0 & \text{else} \end{cases} \end{aligned} \tag{132}$$

and

$$\mathbf{C}_{3Y}[(i-1)p^2 + (j-1)p + k] = C_{3y}(i-j, k-j), \quad i, j, k = 1, ..., p \tag{133}$$

Hence,

$$S_{3m}\mathbf{C}_{3Y}(i) = C_{3y}(m-p, i-p), \quad i = 1, ..., p-1 \tag{134}$$

and, for $i = 1, ..., p-1$,

$$\begin{aligned} & S_{3m}\Phi_3\mathbf{C}_{3Y}(i) \\ & \quad = \sum_{j=1}^{p}\sum_{k=1}^{p}\sum_{l=1}^{p} S_{3m}\Phi_3[i, (j-1)p^2 + (k-1)p + l]C_{3y}(j-k, l-k) \\ & \quad = \sum_{k=1}^{p} \alpha(k)C_{3y}(m+1-k, i+1-k) \\ & \quad = -\sum_{k=1}^{p} a(k)C_{3y}(m+k-p, i+k-p) \end{aligned} \tag{135}$$

where we have let $k \to p+1-k$ and used the definition of $\alpha(k)$ to obtain the last equality.

Using (134) and (135) in (131), we obtain

$$\begin{aligned} \beta_{3m}(i) &= \sum_{k=0}^{p} a(k)C_{3y}(m-p+k, i-p+k) \\ &= \sum_{k=0}^{p} a(k)C_{3y}(m-i, p-i-k) \\ &= \gamma_{3w}\sum_{k=0}^{p}\sum_{j=1}^{\infty} a(k)h(j)h(j+m-i)h(j+p-i-k) \\ &= \gamma_{3w}\sum_{j=1}^{\infty} h(j)h(j+m-i)\sum_{k=0}^{p} a(k)h(j+p-i-k) \\ &= \gamma_{3w}\sum_{j=0}^{\infty} h(j)h(j+m-i)b(j+p-i) \\ &= \gamma_{3w}\sum_{k=0}^{q} b(k)h(k+i-p)h(k+m-p) \end{aligned}$$

where we have used (16) and (96), and let $j+p-i \to k$ to obtain the last equality. Let $m = \bar{m} = p - q + 1 \geq 1$ in the last equation. Then,

$$\begin{aligned}
\beta_{3\bar{m}}(i) &= \gamma_{3w} \sum_{k=0}^{q} b(k)h(k+i-p)h(k-q+1) \\
&= \gamma_{3w} b(q)h(1)h(i+1-\bar{m}) = b(q)h(i+1-\bar{m})
\end{aligned}$$

where we have used $h(1) = 1$. The last equation leads to

$$h(i) = \frac{\beta_{3m}(i+\bar{m}-1)}{\beta_{3m}(\bar{m})}, \quad i = 1, ..., q-1 = p - \bar{m} \tag{136}$$

which establishes (95), where the scalar $\alpha^{-1} = \beta_{3m}(\bar{m}) = b(q)$. $\square$

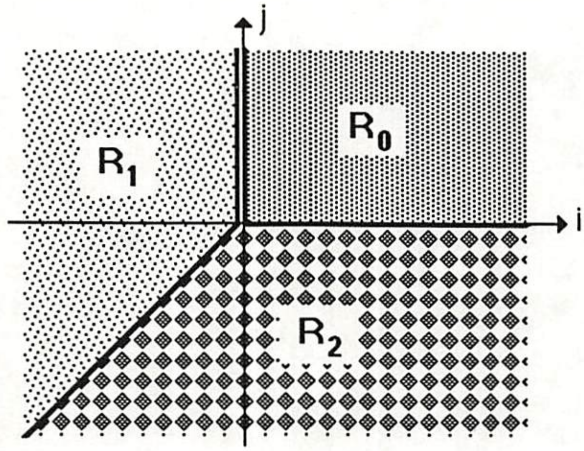


Figure 1

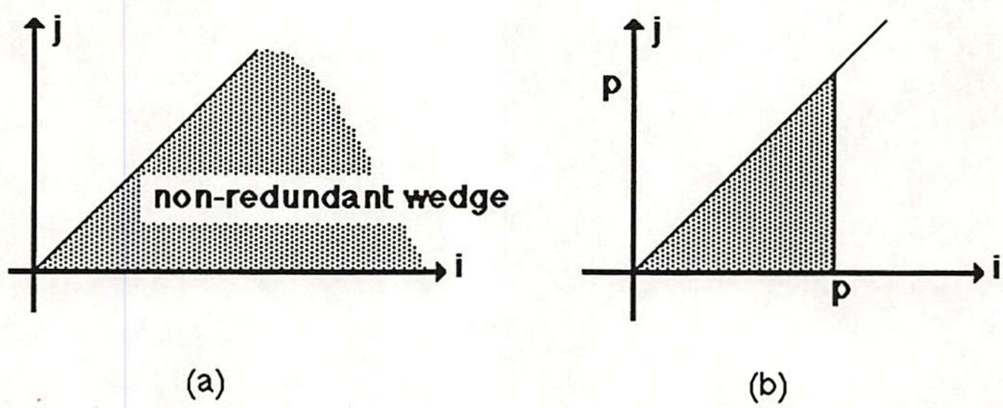


Figure 3

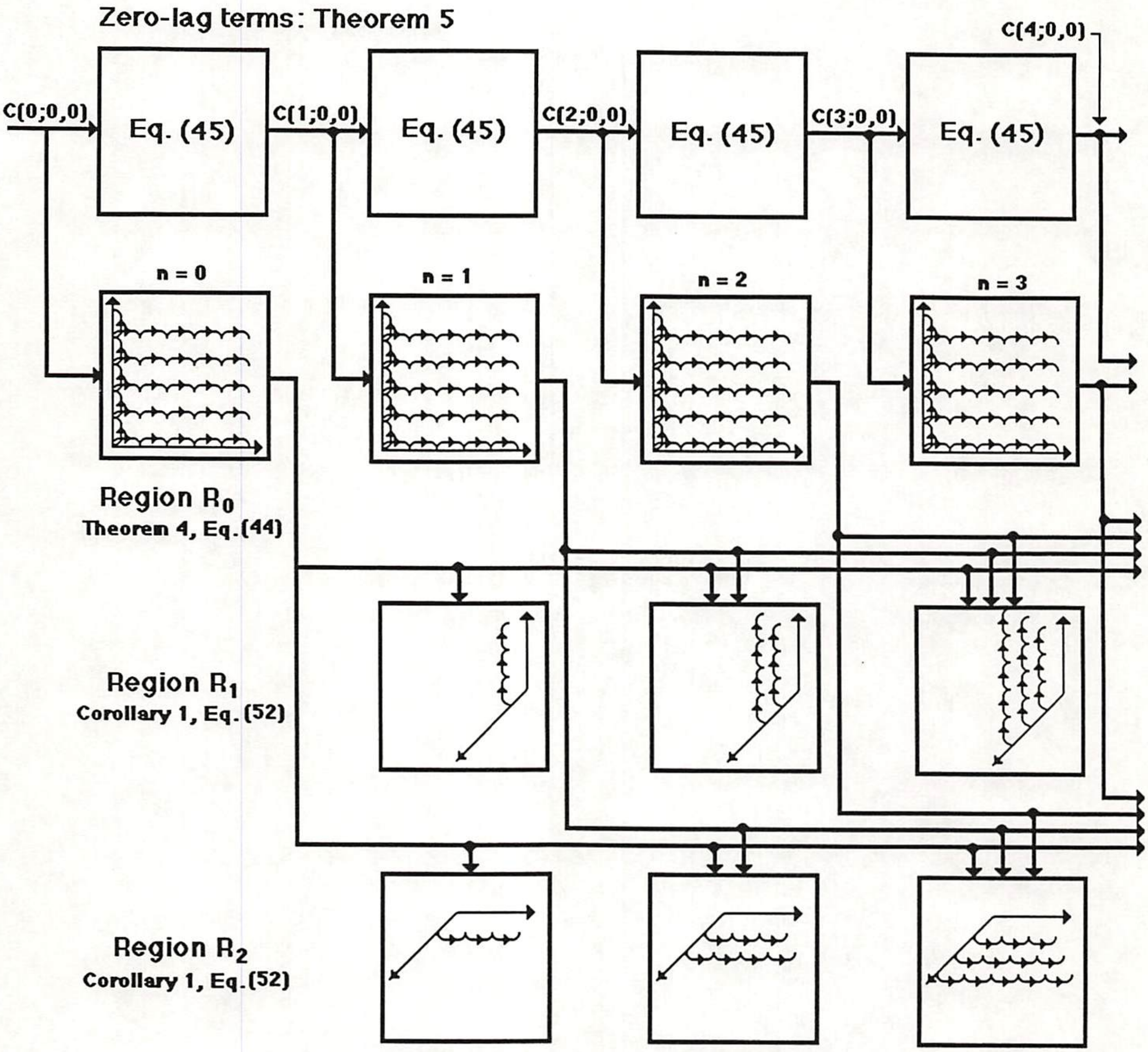

Zero-lag terms: Theorem 5
C(0;0,0)
Eq. (45)
C(1;0,0)
Eq. (45)
C(2;0,0)
Eq. (45)
C(3;0,0)
Eq. (45)
C(4;0,0)
n = 0
n = 1
n = 2
n = 3
Region R0
Theorem 4, Eq.(44)
Region R1
Corollary 1, Eq.(52)
Region R2
Corollary 1, Eq.(52)

Figure 2

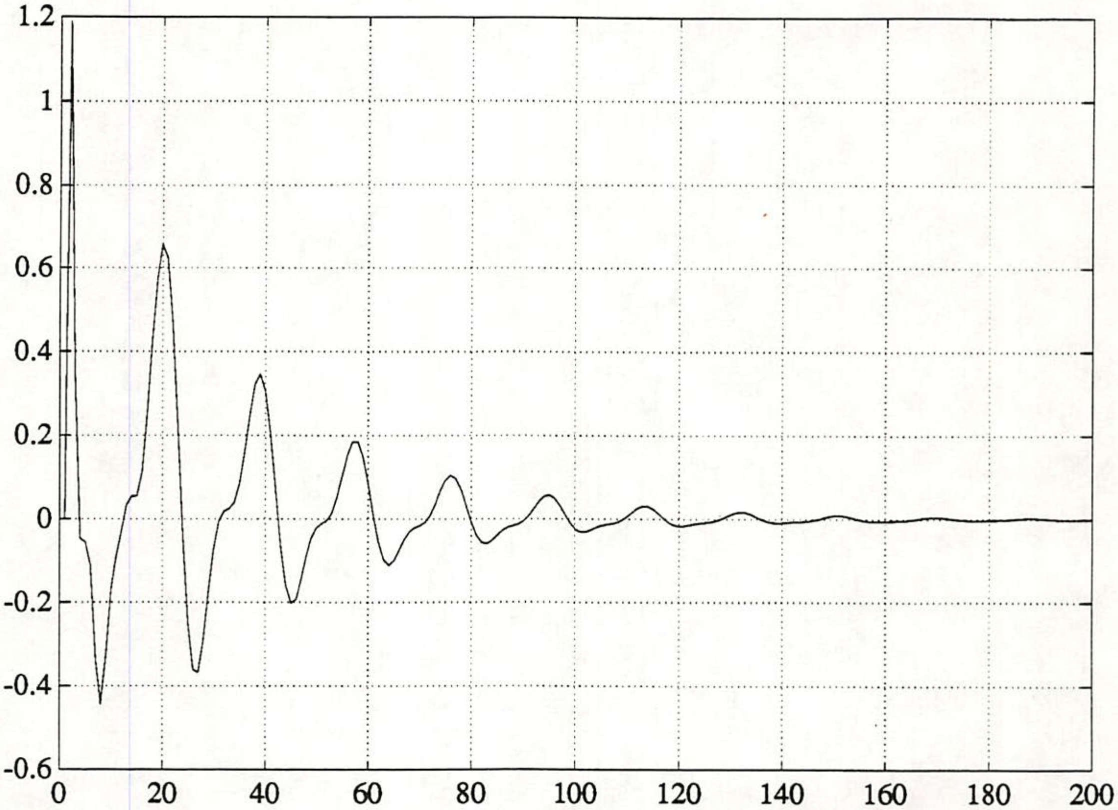


Figure 4a. Impulse response of a tenth-order airgun wavelet

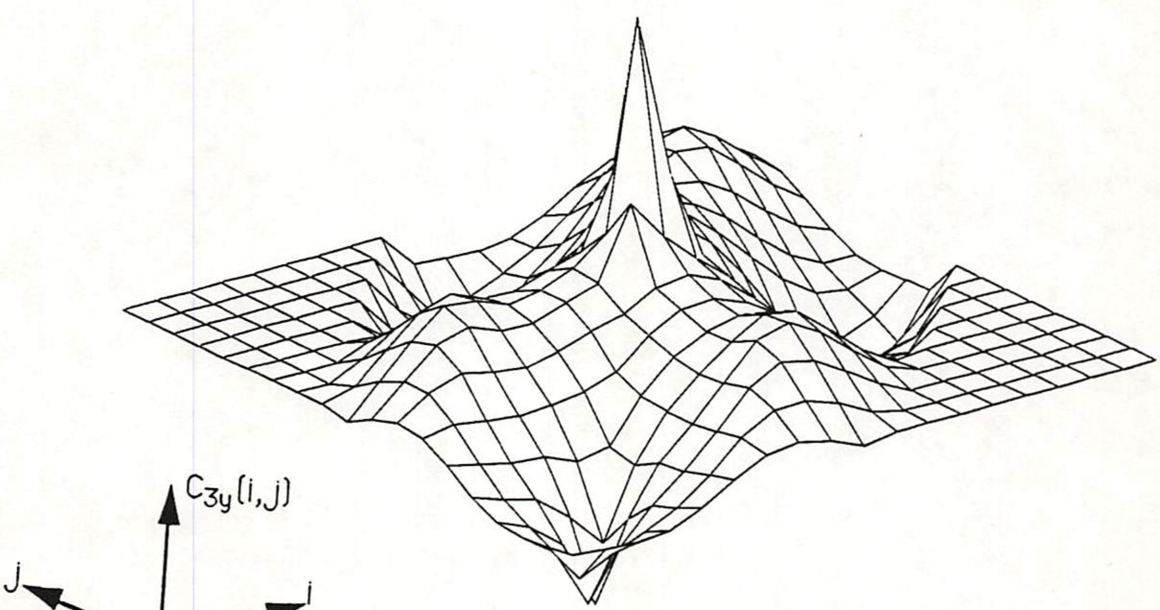
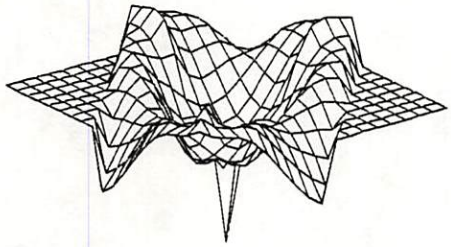
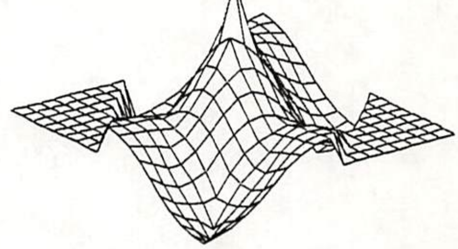


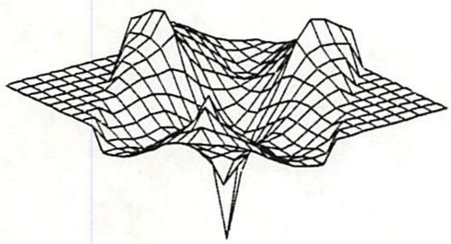
Figure 4b. Third-order output cumulant of a tenth-order LTI SISO system. Cumulant lags, C (i,j), -10 < i,j < 10, are shown.



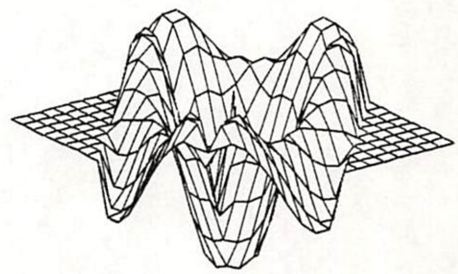
(5a)



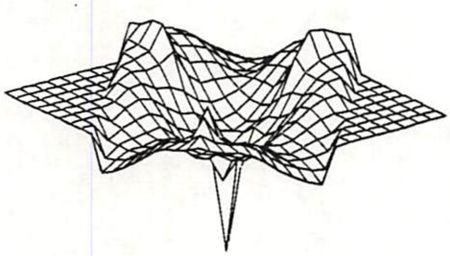
(5b)



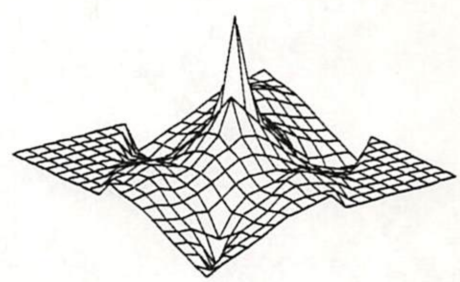
(5c)



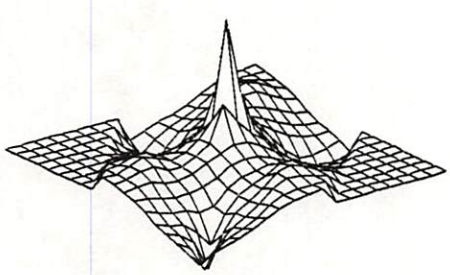
(5d)



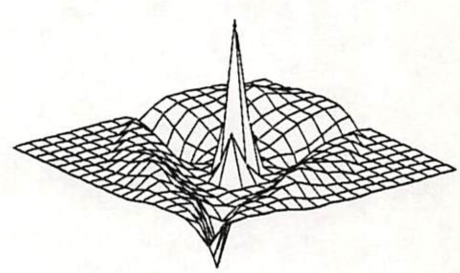
(5e)



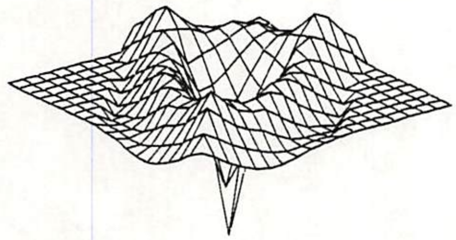
(5f)



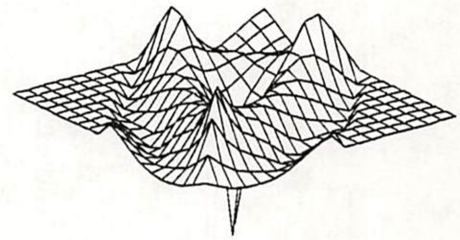
(5g)



(5h)



(5i)



(5j)

Figure 5. Fourth-order output cumulant of a tenth-order LTI SISO system. Cumulant lags, C (m,n,k), -10 < m,n <10, are shown in Figs. (5a)-(5j), for k=0,1,...,9.

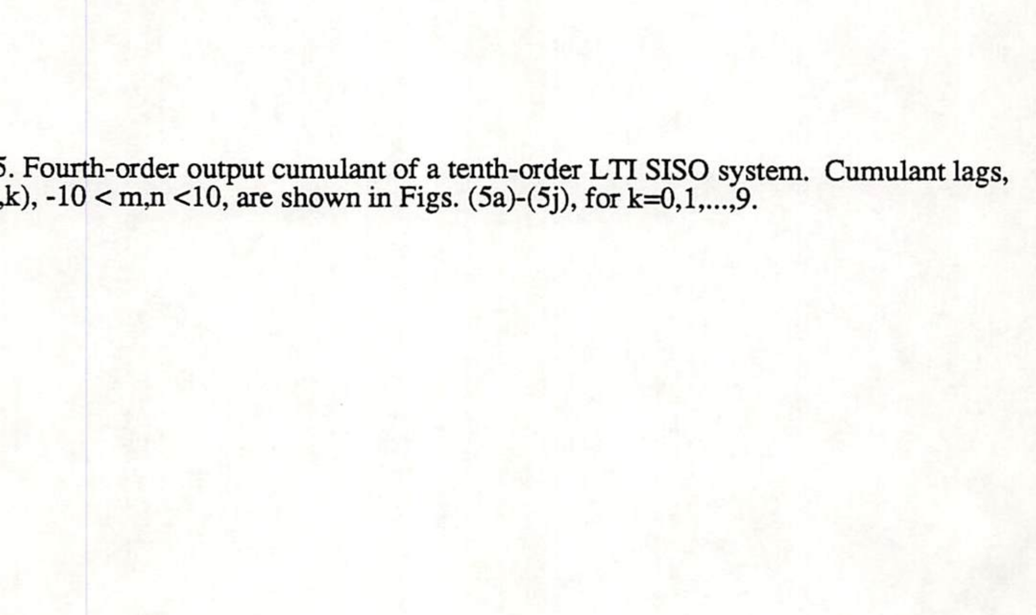


Figure 6. Third-order output cumulant of a tenth-order non-statioanry scalar process