NFORMATION AND CONTROL 38, 179–212 (1978)

On the Algebraic Structure of Certain Partially Observable Finite-State Markov Processes

ALAN S. WILLSKY*

Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

We consider a class of nonlinear estimation problems possessing certain algebraic properties, and we exploit these properties in order to study the computational complexity of nonlinear estimation algorithms. Specifically, we define a class of finite-state Markov processes evolving on finite groups and consider noisy observations of these processes. By introducing some concepts from the theory of representations of finite groups, we are able to define a pair of "dual" filtering algorithms. We then study several specific classes of groups in detail, and, by developing a generalization of the fast Fourier transform algorithm, we derive an efficient nonlinear filtering algorithm. A continuous-time version of these ideas is developed for cyclic groups.

1. INTRODUCTION

The nonlinear filtering problem has proven to be an extremely difficult one. When the filtering problem is described by a set of stochastic differential equations, the optimal nonlinear filter in general requires the solution of a stochastic partial differential equation for the conditional density or the solution of an infinite set of stochastic differential "moment" equations (Jazwinski, 1970). In the case of discrete-time partially observable finite-state Markov processes (POFSMP), the solution is conceptually far simpler, as the conditional distribution can be computed sequentially via straightforward finite-dimensional difference equations (Astrom, 1965). However, even in this conceptually simple case, the nonlinear filtering problem can be computationally nontrivial. Specifically, we note that if we are considering an n-state POFSMP, a straightforward implementation of the conditional distribution update equations requires $O(n^2)$ multiplications. (see Astrom (1965) and the following sections of this paper). For n of reasonable size this becomes an extremely demanding computational task.

* Associate Professor of Electrical Engineering and Computer Science, M.I.T. This work was supported in part by NSF under Grants GK-42090 and GK-41647 and in part by the Science Research Council of Great Britain under a Senior Visiting Fellowship to the Dept. of Computing and Control, Imperial College of Science and Technology, London.

0019-9958/78/0382-0179$02.00/0

In this paper we investigate the computational complexity of the nonlinear filtering problem for a class of POFSMPs possessing particular algebraic properties. We have been motivated by the following observation. Let x and y be independent random variables taking values in the cyclic group $Z_n = \{0, 1,..., n-1\}$ with addition defined modulo n. Let $p_x(j)$ and $p_y(j)$ denote the probabilities that $x = j$ and $y = j$, respectively, and define

$$w = (x + y) \bmod n$$

Then the probability distribution p_w is given by the cyclic convolution

$$p_w(j) = \sum_{k=0}^{n-1} p_x(j-k)\, p_y(k) \triangleq p_x * p_y \tag{1}$$

where all integers in the summation are to be interpreted modulo n. Note that the straightforward calculation of p_w from (1) requires n^2 multiplications. However, it is well known (Oppenheim and Schafer, 1975; Nicholson, 1971; Good, 1958), that, with the aid of a fast Fourier transform (FFT) algorithm, one can reduce the computational load to $0(n \log n)$ multiplications (assuming n is a highly composite number).

In this paper we will generalize these ideas to certain random processes defined over finite groups. Motivated by the comments made previously concerning the cyclic group case, we are led to utilize the natural generalization of (1) to obtain one form for the nonlinear filtering equations. A second form for the filtering equations can be found by examining more closely the use of the FFT in the Z_n case. Specifically, by making use of results on harmonic analysis on finite groups, we obtain a "dual" form for the filtering equations. That is, with every finite group G we associate a "dual" object, G^*, consisting of a complete set of irreducible representations of G over the complex numbers, C. We then can regard G and G^* as bases for the set of all functions from G into C, and hence any probability distribution p on G can be represented by its components with respect to either basis. We then find that the optimal estimation procedure can be broken into two stages. The first, called the "diffusion update," consists of the incorporation of the new randomness introduced into the system between measurements. This stage leads to what can be readily identified as a convolution with respect to the G-basis and a pointwise product with respect to the G^*-basis. The second stage, the "measurement update," involves the incorporation of the information contained in the latest measurement. The major part of the computation in this stage consists of a pointwise product with respect to the G-basis and a convolution product with respect to the G^*-basis (this is, in fact, how we will define convolution with respect to the G^* basis).

Our use of representation theory in the study of stochastic processes on groups is very much in the spirit of the work of Grenander (1963), who utilized Fourier analysis to study the problem of the multiplication of independent random variables on a compact group. Depeyrot (1968, 1971) and Paz (1971) have also studied a "dual automaton" approach to stochastic automata (without measurements). In our work, we extend the analyses of these authors in order to study the nonlinear filtering problem, and we obtain a picture of the duality between diffusion and measurement updates and the two formulations of the nonlinear filtering equations. In this manner we are able to uncover some of the key computational issues in the filtering problem. In particular, our formulation of the dual filtering algorithms has led to a generalization of the FFT to the fast convolution of functions over certain nonabelian groups. One possible application of our results is in the design of efficient decoders for certain codes defined over groups (see Willsky, 1973, 1975; Chizeck, 1976).

In Section 2 we define the problem of interest and recall some results from the theory of group representations. Section 3 contains the description of the two filtering algorithms and a discussion of their structure and computational complexity. The issue of computational complexity is taken up in more detail in Sections 4–6, in which we consider several specific examples. The problems of smoothing and prediction are briefly addressed in Section 7, while in Section 8 we discuss other probabilistic issues that can be considered within our framework. Finally in Section 9 we discuss some of the possible extensions and uses for these results.

2. Definitions and Background

Let U, X, and Y be finite groups, and let $a\colon X \to X$, $b\colon U \to X$, and $c\colon X \to Y$ be group homomorphisms. A *random finite group homomorphic sequential system* (RFGHSS) is a system of the form

$$x(k+1) = b[u(k)]\, a[x(k)], \qquad k \geqslant 0, \tag{2}$$

$$y(k) = v(k)\, c[x(k)], \qquad k \geqslant 1, \tag{3}$$

where $\{u(k)\}$ and $\{v(k)\}$ are sequences of independent random variables in U and Y, respectively, independent of each other and of $x(0)$. We let $\eta(k)$, $\xi(k)$, and $\rho(0)$ denote the probability distributions for $u(k)$, $v(k)^{-1}$, and $x(0)$, respectively. We regard these distributions as functions on the respective groups. For example, $\eta(k)_\mu$ is the probability that $u(k) = \mu$.

Let $\rho(k \mid l)$ denote the conditional probability distribution for $x(k)$ given the measurements $y(1),\ldots, y(k)$ (here $\rho(0 \mid 0) \triangleq \rho(0)$). The nonlinear filtering problem consists of finding an algorithm for the sequential computation of $\rho(k+1 \mid k)$ and $\rho(k \mid k)$. We also consider the prediction problem (compute $\rho(k \mid l)$, $l < k$) and the smoothing problem ($l > k$).

In order to provide the proper setting for the development of our results, we recall some definitions from group representation theory and perform several computations. Let X be a finite group with cardinality n, and let $C[X]$ denote the n-dimensional complex group algebra of all complex-valued functions on X. We represent any $\rho \in C[X]$ as a formal sum

$$\rho = \sum_{g \in x} \rho_g \cdot g, \tag{4}$$

where ρ_g is the value of ρ at g. The operations of pointwise addition and scalar multiplication of functions are defined in the usual way, while the operation of multiplication in $C[X]$ is *function convolution*

$$\begin{aligned} \rho * \eta &= \sum_{g,h \in X} \rho_g \eta_h \cdot gh = \sum_{t \in X} \gamma_t \cdot t, \\ \gamma_t &= \sum_{g \in X} \alpha_g \beta_{g^{-1}t} = \sum_{g \in X} \alpha_{tg^{-1}} \beta_g \,. \end{aligned} \tag{5}$$

We can regard X as a subset of and, in fact, a basis for $C[X]$ with the obvious identification of g with $1 \cdot g$. The convolution of g with any element of $C[X]$ is particularly simple:

$$(\rho * g) = \sum_{h \in X} \rho_h \cdot hg = \sum_{t \in X} \rho_{tg^{-1}} \cdot t. \tag{6}$$

In this case the action of g simply permutes the components of ρ. In addition, $C[X]$ can be made into a commutative algebra independent of the structure of X be endowing it with the *pointwise product*

$$(\rho\eta)_g = \rho_g \eta_g \,. \tag{7}$$

We note that the use of formal series such as (4) to represent probability distributions for finite-state Markov processes is very much in the spirit of the work of Fliess (1972, 1976), who has developed an extensive system-theoretic methodology in terms of formal power series. Fliess' study of finite-state Markov processes involves the examination of realizability conditions — i.e. when a particular finite state stochastic process can be thought of as being derived from a finite-state Markov process. We refer the reader to the references for details.

Let $f\colon G \to H$ be a group homomorphism. We define two maps between the group algebras of G and H. The first of these, also denoted by f, is called the *extension* of f. It maps $C[G]$ into $C[H]$ via

$$f\Big(\sum_{g \in G} \rho_g \cdot g\Big) = \sum_{g \in G} \rho_g \cdot f(g) = \sum_{h \in H} \Big(\sum_{g \in f^{-1}(h)} \rho_g \Big) \cdot h. \tag{8}$$

It is easily seen that the definition given in (8) simply involves the extension of a function defined on a basis (i.e., G) to a linear map on all of $C[G]$. It is also clear that the extended map is a convolution homomorphism if the original function on G is a group homomorphism.

The second map, $\hat{f} : C[H] \to C[G]$, is called the *pullback* of f

$$\hat{f}\Big(\sum_{h\in H} \eta_h \cdot h\Big) = \sum_{h\in H} \eta_h \cdot \Big(\sum_{g\in f^{-1}(h)} g\Big). \tag{9}$$

This is clearly a linear map, and it can also be readily shown that it is a pointwise homomorphism. This latter statement is true independent of any assumptions on f. As we will see, however, the structure of f can be useful in computing $\hat{f}$.

We now introduce the Fourier transform of a function on a finite group (Curtis and Reiner, 1966). For our purposes, we define an n-dimensional representation of a finite group X as a homomorphism T of X into the group of invertible $n \times n$ complex-valued matrices. Two representations T and V are said to be *equivalent* if there exists an invertible $n \times n$ matrix P such that

$$PT(g)\,P^{-1} = V(g), \qquad g \in X.$$

By the *direct sum* of two representations T and V we mean the representation

$$T \oplus V(g) = \operatorname{diag}(T(g),\ V(g)) = \begin{bmatrix} T(g) & 0 \\ 0 & V(g) \end{bmatrix}.$$

A representation is said to be *reducible* if it is equivalent to the direct sum of two other representations. Otherwise, it is *irreducible.*

One of the most fundamental results of group representation theory is the following (Curtis and Reiner, 1966): Suppose X has cardinality n; then there exists a *complete set* of irreducible representations $T^1,\ldots, T^s$ such that

(i) Any representation is equivalent to a direct sum of several of the T^i (where some T^i may be repeated). Thus any irreducible representation is equivalent to some T^i.

(ii) If we let $z_i = \dim(T^i)$, we have

$$\sum_{i=1}^{s} z_i^2 = n. \tag{10}$$

(iii) Let t^i_{jk} denote the element in the jth row and kth column of T^i. These functions satisfy the *orthogonality relations*

$$\frac{z_p}{n} \sum_{g\in X} T^i(g)\, t^p_{lm}(g^{-1}) = E^p_{ml}\delta_{ip}\,, \tag{11}$$

where δ_{ip} is the Kronecker delta, and E^p_{ml} is the $z_p x z_p$ matrix whose (i, j) element is $\delta_{jm}\delta_{kl}$.

(iv) From (10) and (11), we see that the t^i_{jk} form a basis for $C[X]$, and, if $\phi \in C[X]$, we can compute

$$\phi_g = \sum_{i=1}^{s} \sum_{j,k=1}^{z_i} c^i_{jk}(\phi)\, t^i_{jk}(g), \tag{12}$$

where c^i_{jk} is the (j, k) element of the ith *transform matrix*

$$C^i(\phi) = \frac{z_i}{n} \sum_{g \in X} \phi_g [T^i(g^{-1})]'. \tag{13}$$

(v) Without loss of generality, we can take $T^1 \equiv 1$.

Thus we have two representations of functions in $C[X]$. The first of these, (4), can be regarded as an expansion of the function with respect to the set of basis functions $\{g\}$. The expansion (12) is in terms of the basis functions $\{t^i_{jk}(g)\}$. We have seen in Eqs. (5)–(9) how the coordinates with respect to the first basis are mapped under the various operations of interest in this study. We will now see how the coordinates defined by the transform matrices (13) are mapped. Let ρ and η be elements of $C[X]$; then a straightforward calculation (Curtis and Reiner, 1966) yields

$$C^i(\rho * \eta) = \frac{n}{z_i}\, C^i(\rho)\, C^i(\eta). \tag{14}$$

As we shall see, it is this relation that will provide the basis for the computational savings that can be achieved in the efficient solution of our problem.

As a special case of (14), consider the case in which ρ is arbitrary and $\eta = g$. In this case

$$C^i(\eta) = \frac{z_i}{n}\, [T^i(g^{-1})]'. \tag{15}$$

As pointed out earlier, the product $\rho * g$ involves a permutation of the components of ρ. In the transform domain, we must compute the products (14), where the $C^i(\eta)$ are as in (15). Thus the calculation of the transform matrices for $\rho * \eta$ involves a number of multiplications, and the complexity of the computation depends upon the structure of the T^i. For example, in many cases one can take the T^i to be *monomial representations* (Curtis and Reiner, 1966). That is, for any i and any $g \in X$, there is only one nonzero element in any column or row of $T^i(g)$. In this case it is easily seen that the calculation of $C^i(\rho * \eta)$, $i = 1,..., s$, involves at most n multiplications.

There are three remaining calculations that must be performed to determine the transform counterparts of (7)–(9). We have placed these somewhat involved computations in Sections A.1 and A.2 of the Appendix, and for simplicity we adopt the notation

$$L_i[C(\rho), C(\psi)] \triangleq C^i(\rho\psi), \tag{16}$$

$$f_i[C(\rho)] \triangleq C^i[f(\rho)], \tag{17}$$

$$\hat{f}_i[C(\rho)] \triangleq C^i[\hat{f}(\rho)]. \tag{18}$$

Referring to our discussion in Section 1, we see that (16) can be interpreted as a convolution with respect to the set of basis functions $\{t^i_{jk}\}$. This observation will lead to the formulation of our "dual" filtering algorithm.

3. The Filtering Algorithm

Regarding all of the relevant probability distributions as elements of the appropriate group algebras, we obtain the following

Proposition 1. *Consider the estimation problem described in Section 2. We gave the following filtering algorithm:*

Diffusion update

$$\rho(k+1 \mid k) = b[\eta(k)] * a[\rho(k \mid k)]. \tag{19}$$

Measurement update

$$\lambda(k) = \xi(k) * y(k), \qquad \mu(k) = \hat{c}[\lambda(k)], \tag{20}$$

$$\gamma(k \mid k) = \mu(k)\, \rho(k \mid k-1), \tag{21}$$

$$N(k \mid k) = \sum_{g \in X} \gamma(k \mid k)_g, \tag{22}$$

$$\rho(k \mid k) = \frac{\gamma(k \mid k)}{N(k \mid k)}. \tag{23}$$

The proof of this result is a straightforward application of Bayes' rule (see Section A.3). With the aid of the calculations made in the preceding section, we obtain a "dual" filtering algorithm.

Corollary. An alternative form for the filtering equations is

Diffusion update

$$C^i(\rho(k+1 \mid k)) = (n/z_i)\, b_i[C(\eta(k))]\, a_i[C(\rho(k \mid k))]. \tag{24}$$

Measurement update

$$C^i(\lambda(k)) = C^i(\xi(k))\, T^i[y(k)^{-1}],\ C^i(\mu(k)) = \hat{c}_i[C(\lambda(k))], \tag{25}$$

$$C^i(\gamma(k \mid k)) = L_i[C(\mu(k)),\, C(\rho(k \mid k-1))], \tag{26}$$

$$C^i(\rho(k \mid k)) = \frac{C^i(\gamma(k \mid k))}{nC^i(\gamma(k \mid k))}\,. \tag{27}$$

Let us now take a look at the two algorithms step by step:

1. *Diffusion update*

A. *Action of the homomorphism a.* In the distribution domain, the necessary calculation is

$$a[\rho(k \mid k)]_g = \sum_{h \in a^{-1}(g)} \rho(k \mid k)_h\,. \tag{28}$$

Thus, if a is an isomorphism, (28) is simply a permutation of the coefficients of ρ. If ker a has t elements, then $a(\rho)$ has only (n/t) nonzero elements, and for each we must perform $(t-1)$ additions (each element in $a(x)$ has precisely t preimage points), yielding a total of $n(t-1)/t$ additions.

The effect of the homomorphism a in the transform domain is considered in Section A.2 (see equations (81)–(85)). Essentially the calculation of $a_i(\rho)$ involves only similarity transformations of the $C^i(\rho)$ and of direct sums of the $C^i(\rho)$. The calculations are particularly simple if the T^i and the S^i (a complete set of irreducible representations of $a(X)$) are monomial, or if a is an isomorphism. Our examples in Sections 4–6 will indicate the simplicity of this step.

B. *Effect of the noise on the state.* We want to argue that, without loss of generality, we can assume that $U = X$ and $b =$ identity. This basically comes from the argument that we can just as easily regard $b[u(k)]$ as the basic driving noise. Equivalently, if the $\eta(k)$ are assumed to be known a priori, we can precompute $b[\eta(k)]$ or $b_i[C(\eta(k))]$. Assuming that $U = X$, $b =$ identity, equations (19) and (24) become

$$\rho(k+1 \mid k) = \eta(k) * a[\rho(k \mid k)], \tag{29}$$

$$C^i(\rho(k+1 \mid k)) = (n/z_i)\, C^i(\eta(k))\, a_i[C(\rho(k \mid k))]. \tag{30}$$

To get an idea of the complexity of the convolution in (29), let us assume that $\eta(k)$ is the distribution of $b[u(k)]$, where $u \in U$, and card $[b(U)] = m$. In this case, η has only m nonzero elements. Letting $M = a(X)$ with card(ker a) $= t$, we find that a straightforward calculation of (29) requires $(mn)/t$ multiplications. To compute the number of additions, let

$$\text{card}[M \cap b(U)] = l_1\,, \qquad \text{card}[b(U)M] = l_2\,.$$

Then $l_2(l_1 - 1)$ additions are required. Note that if our system (2) is *controllable* (i.e., if every state $x \in X$ can be reached from any other by an appropriate sequence of inputs), then $b(U)M = X$ and $l_2 = n$ (Brockett and Willsky, 1972).

In the transform domain, we calculate the s matrix products (30), which require at most

$$\sum_{i=1}^{s} z_i^3 \text{ multiplications and } \sum_{i=1} z_i^3 - n \text{ additions.}$$

The use of the transform domain for this calculation can be quite efficient in certain cases. For example, if $m = n$ and $t = 1$, the distribution computation requires n^2 multiplications and $(n^2 - n)$ additions. These numbers can be much larger than the corresponding numbers in the transform domain. For example, if X is Abelian, then $s = n$ and $z_i = 1$ for all i (Curtis and Reiner, 1966), and the transform calculation consists of n multiplications and no additions. As we will see, it is Step B that is the most complex part of the diffusion update. It is here that the utility of fast transform methods will be most apparent.

2. *Measurement Update*

A. *Effect of the measurement: calculation of* $\lambda(k)$ *and* $C^i(\lambda(k))$. As described in Section 2, (20) consists simply of a permutation of the elements of ξ. On the other hand, (25) consists of a number of matrix multiplications. As pointed out earlier, if the T^i are monomial representations, then we require at most n multiplications to compute (25) for all s. As we shall see in the examples described in subsequent sections, the transform calculation (25) can often be simplified even more.

B. *Effect of the pullback map* $\hat{c}$. In the distribution domain, we must compute

$$\hat{c}(\lambda(k))_g = \lambda(k)_{c(g)} \,.$$

If c is an isomorphism onto a subgroup of Y, $\hat{c}$ simply permutes the elements of the restriction of λ to $c(X)$. If c has a nontrivial kernel, a number of components of $\hat{c}(\lambda)$ will be equal.

The relevant equations in the transform domain are (86)–(92), where (89) and (92) represent the required on-line calculations (note that (89) is nontrivial only if c is not surjective—i.e., only if the range of the noisy measurement $v(k)c[x(k)]$ is larger than the range of the noise-free measurement $c[x(k)]$). Again, the required calculations, involving linear combinations and similarity transformations, are often extremely simple. We refer the reader to the examples to see what these calculations actually entail.

C. *The numerator of Bayes' rule.* The next step in the distribution domain is the pointwise product of $\hat{c}[\lambda(k)]$ and $\rho(k \mid k - 1)$. This involves n multiplications and no additions. In the transform domain, referring to (26), one must compute the transform matrix of the pointwise product of two elements of $C[X]$. The relevant develoment is in Section A.1 of the Appendix. In general, the calculation in the transform domain for this step is far more complex than in the distribution domain. However, this calculation does display structure that can be exploited. Specifically, for the diffusion update, Step B consists of a convolution in the distribution domain and a "pointwise product" of transform matrices in the transform domain. On the other hand, for the measurement update, Step C consists of a pointwise product in the distribution domain and (78) in the transform domain. As we will see in the examples, in at least some cases (78) turns out to be a convolution.

D. *Normalization.* In the distribution domain, (22), (23) requires $(n-1)$ additions and n divisions, while in the distribution domain, we need one multiplication and $(n-1)$ divisions.

4. An Example: The Cycle Group Z_n

Consider the cyclic group Z_n, which we interchangeably identify with the integers and with the set $\{\tau^k\}$, where integer addition is defined modulo n. We adopt the notation

$$\phi = \sum_{k=0}^{n-1} \phi_k \tau^k$$

for elements of $C[Z_n]$. All of the irreducible representations are one-dimensional, and a complete set of these is given by

$$T^i(\tau) = e^{j2\pi i/n} \triangleq \gamma^i, \qquad i = 0, \ldots, n-1$$

(here $j = (-1)^{1/2}$). In this case, the transform pair (12), (13) are the usual finite Fourier series equations

$$C^i(\phi) = \frac{1}{n} \sum_{m=0}^{n-1} \phi_m \gamma^{-im}, \qquad \phi_m = \sum_{i=0}^{n-1} C^i(\phi) \gamma^{im}. \tag{31}$$

Consider the most general RFGHSS with $U = X = Y = Z_n$ $(a, c \in Z_n)$

$$x(k+1) = ax(k) + u(k), \qquad y(k) = cx(k) + v(k). \tag{32}$$

1. *Diffusion Update*

A. *Effect of multiplication by a.* Let $\phi \in C[Z_n]$ and let $d = gcd(a, n)$. Then aZ_n has cardinality n/d, and a straightforward calculation yields

$$(a\phi)_{ka} = \sum_{j=0}^{d-1} \phi_{k+(jn/d)} \,. \tag{33}$$

Also, from (31) we have

$$a_i[C(\phi)] = C^{ai}(\phi). \tag{34}$$

Thus in the distribution domain, (33) requires $n(d - 1)/d$ additions, while (34) is simply a reshuffling of the Fourier coefficients. Note that (33), (34) are permutations of the elements of ϕ and $C(\phi)$ if and only if a is invertible—i.e., $d = 1$.

B. *Effect of the convolution.* Let $\phi, \psi \in C[Z_n]$. The convolution product is given by

$$(\phi * \psi)_k = \sum_{m=0}^{n-1} \phi_m \psi_{k-m}\,, \tag{35}$$

which requires n^2 multiplications and $n(n - 1)$ additions. The transform

$$C^i(\phi * \psi) = nC^i(\phi)\, C^i(\psi)$$

requires n multiplications.

2. *Measurement Update*

A. *Effect of the measurement.* Let $\xi \in C[Z_n]$ and consider $\xi * \tau^k$. Application of the results of the preceding section yields

$$(\xi * \tau^k)_m = \xi_{m-k}\,, \qquad C^i(\xi * \tau^k) = \gamma^{-ik} C^i(\xi).$$

B. *Effect of the pullback map.* Let $\phi \in C[Z_n]$. Then using (31) and the definition of $\hat{c}$, we obtain

$$(\hat{c}\phi)_m = \phi_{cm}\,. \tag{36}$$

Also, if we let $f = gcd(c, n)$, we find that the only nonzero Fourier coefficients of $\hat{c}\phi$ are $\hat{c}_{lc}[C(\phi)]$, $l = 0, 1,..., n/f - 1$, where

$$\hat{c}_{lc}[C(\phi)] = \sum_{r=0}^{f-1} C^{l+(rn/f)}(\phi). \tag{37}$$

Thus (36) is a simple reshuffling, while (37) requires $n(f - 1)/f$ additions (note that here the subgroup matrices are trivial). Also (36), (37) are permutations if and only if $f = 1$.

C. *Effect of the pointwise product.* Let $\phi, \psi \in C[Z_n]$. Then the pointwise product $\phi\psi$ requires n multiplications, while in the transform domain we obtain the convolution

$$C^i(\phi\psi) = \sum_{r=0}^{n-1} C^r(\phi)\, C^{i-r}(\psi), \tag{38}$$

which requires n^2 multiplications and $n(n-1)$ additions. This follows easily from the results of Section A.1 when we observe that

$$T^i T^\alpha = T^{i+\alpha} \Rightarrow L^{(i,\alpha,\gamma)} = \delta_{i,\gamma-\alpha}\,.$$

TABLE I

Required Computations for the Two Z_n Filtering Algorithms

Group algebra	Transform
1.A. $(a\rho(k \mid k))_{la} = \sum_{r=0}^{d-1} \rho(k \mid k)_{l+rn/d}$	1.A. $a_i(C(\rho(k \mid k))) = C^{ai}(\rho \mid k))$
B. $\rho(k+1 \mid k)_m = \sum_{r=0}^{n-1} \eta(k)_r(\alpha\rho(k \mid k))_{m-r}$	B. $C^i(\rho(k+1 \mid k)) = nC^i(\eta(k))\, a_i(C(\rho(k \mid k)))$
2.A. $\lambda(k)_m = \xi(k)_{m-y(k)}$	2.A. $C^i(\lambda(k)) = \gamma^{-iy(k)} C^i(\xi(k))$
B. $(\hat{c}(\lambda(k)))_m = \lambda(k)_{cm}$	B. $\hat{c}_{lc}(C(\lambda(k))) = \sum_{r=0}^{f-1} C^{l+rn/f}(\lambda(k))$
C. $\gamma(k \mid k)_m = (\hat{c}(\lambda(k)))_m\, \rho(k \mid k-1)_m$	C. $C^i(\gamma(k \mid k))$ $= \sum_{r=0}^{n-1} \hat{c}_r(C(\lambda(k)))\, C^{i-r}(\rho(k \mid k-1))$
D. $\rho(k \mid k) = \dfrac{\gamma(k \mid k)}{\sum_{m=0}^{n-1} \gamma(k \mid k)_m}$	D. $C^i(\rho(k \mid k)) = \dfrac{C^i(\gamma(k \mid k))}{nC^0(\gamma(k \mid k))}$

In Table I we have summarized the two Z_n filtering algorithms for convenience. In this form the duality between the algorithms becomes quite apparent. Comparing steps 1.A and 2.B, we see the duality of the operations on elements of $C[Z_n]$ and their transforms induced by the extensions and pullbacks of homomorphisms. If one compares 1.B and 2.C, we see that the diffusion update induces a convolution in the group algebra domain and a pointwise product in the transform space, while the measurement update requires a pointwise product of group algebra elements or a convolution of their transforms.[1] Finally, if we examine the computational complexity of either algorithm by itself, we find that all calculations other than the convolutions require $O(n)$ multiplications, while

[1] Note the factor of n in 1.B in the transform domain. If we had moved the factor of z_i/n from (13) to (12), the factor of n would have appeared in 2.C in the group algebra domain. A symmetric picture arises if a square root of z_i/n is included in *both* (12) and (13).

a naive calculation of the convolution requires $O(n^2)$ multiplications. The overall computational burden of the filtering problem can clearly be reduced if we can avoid direct calculation of the convolution by performing the associated pointwise multiplication in the other domain. This can be done in an efficient manner if n is a highly composite number—e.g., if $n = 2^\nu$—in which case one can utilize the fast Fourier transform (FFT) to compute (35) or (38) with $O(n \log n)$ multiplications.

We note that the extension of the results of this section to the case in which X, U, and Y are arbitrary finite abelian groups is straightforward. Recall (Rotman, 1965) that any such group G is a direct product of cyclic groups

$$G = Z_{n_1} \times Z_{n_2} \times \cdots \times Z_{n_k}.$$

In addition, all the irreducible representations are one-dimensional and are simply *products* of the representations of the component cyclic groups. Specifically, for any $m = (m_1, ..., m_k) \in G$, we define T^m via

$$T^m(l) = \gamma^{ml} \triangleq \prod_{i=1}^{k} \gamma_i^{m_i l_i},$$

$$l = (l_1, ..., l_k) \in G, \gamma_i = e^{j2\pi/n_i}, \gamma = (\gamma_1, ..., \gamma_k).$$

Examination of these representations shows that the transform of $\phi \in C[G]$ is just a multidimensional finite Fourier Transform and hence can be computed using the multi-dimensional FFT (Oppenheim and Schafer, 1975). Thus, filtering on arbitrary finite abelian groups can be accomplished efficiently, as we can decrease the complexity of convolutions by utilizing FFTs to turn them into pointwise products (see also Depeyrot, 1974). The general finite abelian group case also exhibits duality properties along the lines of those we have displayed for Z_n, although in the special case we considered, the situation is most striking, since the output group is the same as the state group.

5. A Nonabelian Example: The Dihedral Group D_n

Let D_n denote the group of order $2n$ generated by the two elements α and β, which satisfy the relations

$$\alpha^p = \beta^2 = 1, \qquad \alpha\beta\alpha = \beta$$

(note $Z_n \simeq \langle\alpha\rangle$). For $\phi \in C[D_n]$ we write

$$\phi = \sum_{m=0}^{n-1} \sum_{l=0}^{1} \phi_{ml} \alpha^m \beta^l = \sum_{g \in D_n} \phi_g \cdot g$$

(we will use both notations). If n is odd, there are two inequivalent one-dimensional representations, defined by

$$U^0(g) = 1 \qquad \forall g,$$
$$U^1(\alpha) = 1, \qquad U^1(\beta) = -1.$$

If n is even, we have two additional 1-D representations:

$$U^k(\alpha^l\beta^m) = (-1)^l(-1)^{km}, \qquad k = 2, 3.$$

The remaining irreducible representations, V^i, $i = 1,\ldots, \lfloor(n-1)/2\rfloor$ (here $\lfloor x\rfloor$ = largest integer $\leqslant x$), are two-dimensional and are given by

$$V^r(\alpha^k\beta^l) = \begin{bmatrix} \gamma^{kr} & 0 \\ 0 & \gamma^{-kr} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^l, \qquad \gamma = e^{j2\pi/n}.$$

Let $\phi \in C[D_n]$, and define its transform

$$B^i(\phi) = \frac{1}{2n} \sum_{g\in D_n} \phi_g U^i(g^{-1}),$$

$$D^i(\phi) = \frac{1}{n} \sum_{g\in D_n} \phi_g [V^i(g^{-1})]'.$$

Note also that $\langle\alpha\rangle$ has two distinct cosets, itself and $\langle\alpha\rangle\beta$. Restricting ϕ to each of these, we effectively have defined two elements of $C[Z_n]$. Their transforms are given by

$$\mu_k(\phi) = \frac{1}{n} \sum_{m=0}^{n-1} \phi_{m,0}\gamma^{-mk}, \qquad \nu_k(\phi) = \frac{1}{n} \sum_{m=0}^{n-1} \phi_{m,1}\gamma^{-mk}. \tag{39}$$

A straightforward calculation then yields

$$\begin{aligned} B^0(\phi) &= \tfrac{1}{2}(\mu_0 + \nu_0), \qquad B^1(\phi) = \tfrac{1}{2}(\mu_0 - \nu_0), \\ B^2(\phi) &= \tfrac{1}{2}(\mu_{n/2} + \nu_{n/2}), \qquad B^3(\phi) = \tfrac{1}{2}(\mu_{n/2} - \nu_{n/2}), \end{aligned} \tag{40}$$

$$D^i(\phi) = \begin{bmatrix} \mu_i & \nu_i \\ \nu_{n-i} & \mu_{n-i} \end{bmatrix}. \tag{41}$$

Thus we see that we can devise a fast $C[D_n]$ transform algorithm consisting of two FFT's to compute (39), combined with the identifications (40), (41).

Let us now turn to filtering on D_n. Consider (2), (3) with $U = X = Y = D_n$, b = identity, and a and c defined by

$$a(\alpha) = \alpha^{n-1}, \qquad a(\beta) = \alpha\beta, \tag{42}$$

$$c(\alpha) = \alpha^2, \qquad c(\beta) = \alpha^2\beta, \tag{43}$$

for a given $s \geqslant 0$. We have chosen (42), (43) to illustrate some of the properties of extensions and pullbacks in this setting. The comments concerning the remaining parts of the algorithms hold in general.

1. *Diffusion Update*

A. *Action of the homomorphism a.* Note that a is an isomorphism and that $a^{-1} = a$. Hence, following (81), we compute a complete set of irreducible representations $\{\tilde{U}_i, \tilde{V}_i\}$ from $\{U_i, V_i\}$ by setting $P_i = I$, $\forall i$. Then, from (81), we have

$$\tilde{U}_i = U^i \circ a^{-1} = U^i \circ a.$$

This yields $\tilde{U}^0 = U^0$, $\tilde{U}^1 = U^1$, $\tilde{U}^2 = U^3$, $\tilde{U}^3 = U^2$. Also

$$\tilde{V}^i = V^i \circ a, \qquad V^i = R_i \tilde{V}^i R_i,$$

$$R_i = R_i^{-1} = \begin{bmatrix} 0 & \gamma^{i/2} \\ \gamma^{-i/2} & 0 \end{bmatrix}.$$

We can then apply (84)

$$\begin{aligned} &B^i(a(\rho)) = B^i(\rho), \qquad i = 0, 1, \\ &B^2(a(\rho)) = B^3(\rho), \qquad B^3(a(\rho)) = B^2(\rho), \end{aligned} \tag{44}$$

$$D^i(a(\rho)) = R_i' D^i(\rho) R_i. \tag{45}$$

Thus (44), (45) represents a permutation of the transform matrices, together with the multiplications implied in (45). On the group algebra side, a permutes the elements of ρ. Explicitly

$$a(\rho)_{m,0} = \rho_{n-m,0}, \qquad a(\rho)_{m,1} = \rho_{n-m+1,1}.$$

B. *Effect of the convolution.* Using the multiplication rules in D_n, we have

$$(\phi * \psi)_{m,0} = \sum_{k=0}^{n-1} \phi_{k,0}\psi_{m-k,0} + \sum_{k=0}^{n-1} \phi_{k,1}\psi_{k-m,1}, \tag{46a}$$

$$(\phi * \psi)_{m,1} = \sum_{k=0}^{n-1} \phi_{k,0}\psi_{m-k,1} + \sum_{k=0}^{n-1} \phi_{k,1}\psi_{k-m,0}. \tag{46b}$$

The straightforward calculation of (46) requires $4n^2$ multiplications. However, with the aid of transforms, we can find two faster ways to perform this step. First, recall that in terms of the transform matrices

$$B^i(\phi * \psi) = 2nB^i(\phi)\, B^i(\psi), \tag{47a}$$

$$D^i(\phi * \psi) = nD^i(\phi)\, D^i(\psi). \tag{47b}$$

The calculations in (47a) are scalar, while those in (47b) consist of 2 × 2 matrix multiplications, each of which can be performed in 7 multiplications with the aid of Strassen's algorithm (see Strassen (1969) and Borodin and Munro (1975)). Thus counting up the multiplications, we find that (47) requires $(7n - 3)/2$ (n odd) or $(7n - 6)/2$ (n even) multiplications. This, combined with the fast D_n transform described earlier, leads to an efficient implementation of this step of the filtering algorithm.

We note that one also has a second (not quite as) fast algorithm. Examing (46) we see that these calculations consist of four cyclic convolutions. Using the definitions in (39), we find that

$$\begin{aligned} \mu_r(\phi * \psi) &= n[\mu_r(\phi)\, \mu_r(\phi) + \nu_r(\phi)\, \nu_{n-r}(\psi)], \\ \nu_r(\phi * \psi) &= n[\mu_r(\phi)\, \nu_r(\psi) + \nu_r(\phi)\, \mu_{n-r}(\psi)]. \end{aligned} \tag{48}$$

Equation (48) consists of 4 pointwise products of the transforms of ϕ and ψ restricted to the cosets of Z_n and requires $4n$ multiplications.

2. *Measurement Update*

A. *Effect of the measurement.* Let $\xi \in C[D_n]$ and consider $\xi * y$, where y is the measurement

$$y = \alpha^p \beta^l, \qquad 0 \leqslant p \leqslant n - 1, \quad 0 \leqslant l \leqslant 1.$$

Then, for $l = 0$, we have in the distribution domain

$$(\xi * y)_{m,0} = \xi_{m-p,0}\,, \qquad (\xi * y)_{m,1} = \xi_{m+p,1}\,, \tag{49}$$

while for the transform matrices

$$\begin{aligned} B^i(\xi * y) &= B^i(\xi), & i &= 0, 1, \\ B^i(\xi * y) &= (-1)^p\, B^i(\xi), & i &= 2, 3, \\ D^i(\xi * y) &= D^i(\xi) \begin{bmatrix} \gamma^{-pi} & 0 \\ 0 & \gamma^{pi} \end{bmatrix}. \end{aligned}$$

Also, for the Z_n transform

$$\mu_i(\xi * y) = \gamma^{-pi} \mu_i(\xi), \qquad \nu_i(\xi * y) = \gamma^{pi} \nu_i(\xi). \tag{50}$$

If $l = 1$, we have

$$(\xi * y)_{m,0} = \xi_{m+p,1}, \qquad (\xi * y)_{m,1} = \xi_{m-p,0}, \tag{51}$$

$$B^0(\xi * y) = B^0(\xi), \qquad B^1(\xi * y) = -B^1(\xi),$$

$$B^i(\xi * y) = (-1)^{p+i} B^i(\xi), \qquad i = 2, 3,$$

$$D^i(\xi * y) = D^i(\xi) \begin{bmatrix} 0 & \gamma^{pi} \\ \gamma^{-pi} & 0 \end{bmatrix},$$

$$\mu_i(\xi * y) = \gamma^{-pi}\nu_i(\xi), \qquad \nu_i(\xi * y) = \gamma^{pi}\mu_i(\xi). \tag{52}$$

Note that the calculations required here—permutations and multiplications by γ^{pi}—are quite similar to those required in the Z_n case (see Section 4). Examining (49) and (51) we can see that the noncommutativity of D_n results in differences in the direction $(\pm p)$ of the cyclic permutations induced by the measurement. This is also evident in (50), (52), where we see the difference in the multiplicative factor $(\gamma^{\pm p})$ *and* a reversal of the Z_n transforms when $y = \alpha^p\beta$.

B. *Effect of the pullback map.* In the distribution domain we have

$$\hat{c}(\phi)_{m,0} = \phi_{2m,0}, \qquad \hat{c}(\phi)_{m,1} = \phi_{2m+s,1}.$$

This is a permutation if n is odd but not if n is even. To determine the effect in the transform domain, consider the Z_n transforms and the Z_n result obtained in (37). Let $f = \gcd(2, n)$. The nonzero $\mu_k(\hat{c}(\phi))$ are for $k = 2l$, $l = 0, 1, \ldots, n/f - 1$, with

$$\mu_{2l}(\hat{c}(\phi)) = \sum_{r=0}^{f-1} \mu_{l+rn/f}(\phi). \tag{53}$$

This is simply a permutation of the $\mu_i(\phi)$ if n is odd. Similarly, we find

$$\nu_{2l}(\hat{c}(\phi)) = \sum_{r=0}^{f-1} \nu_{l+rm/f}(\phi)\gamma^{s(l+rm/f)}. \tag{54}$$

Again, if n is odd, (54) simply consists of a multiplication of $\nu_l(\phi)$ by γ^{sl}, followed by a permutation. Equations (53) and (54), together with (40), (41) then give us a method for calculating the transform matrices. The detailed calculations for the case $s = 0$ are given in Willsky (1976) and indicate the essentially trivial nature of this step of the algorithm.

C. *Effect of the pointwise product.* For $\phi, \psi \in C[D_n]$ we have

$$(\phi\psi)_{m,0} = \phi_{m,0}\psi_{m,0}, \qquad (\phi\psi)_{m,1} = \phi_{m,1}\psi_{m,1}.$$

We can easily compute the transform version in terms of Z_n transforms

$$\mu_k(\phi\psi) = \sum_{r=0}^{n-1} \mu_r(\phi)\, \mu_{k-r}(\psi), \qquad \nu_k(\phi\psi) = \sum_{r=0}^{n-1} \nu_r(\phi)\, \nu_{k-r}(\psi). \tag{55}$$

The calculation of the version of (55) in terms of the D_n transform matrices can be done from (55) and (40), (41), or we can directly apply (78), which involves the calculation of the characteristic matrices for D_n. This calculation is carried out in Willsky (1976).

Examining the equation for the two D_n filtering algorithms, we again see a duality, although the noncommutativity of D_n does make the picture more complex than in the Z_n case (however the duality of extensions and pullbacks in group algebra and transform domains would have been far more apparent if we had taken $a = c$). Nevertheless the main point—the duality of "pointwise" and "convolution" products in the distribution and transform domains—is clearly evident. As these are the most complex steps computationally, we see that one can obtain a fast D_n filtering algorithm by utilizing the fast D_n-transform to trade a "convolution" for a "pointwise product."

6. A Generalization: Metacyclic Groups

The dihedral groups are a very simple example of the following

Definition. G is a *metacyclic group* if there exists an $a \in G$ such that

(i) $H = \langle a \rangle \lhd G$,

(ii) G/H is cyclic—i.e., there exists $b \in G$ such that

$$G/H = \langle bH \rangle.$$

Then (following Curtis and Reiner (1966)) we can define several integers:

(i) Let $|G| = n$, $|H| = m$.

(ii) By the normality of H, there exists an integer r such that $b^{-1}ab = a^r$.

(iii) Since bH generates G/H, there exist integers s and t such that $b^s = a^t$.

We also have the relations[2]

$$n = ms, \quad \gcd(m, r) = 1, \qquad m \backslash t(r - 1).$$

[2] Here $x \backslash y$ should be read "the integer x divides the integer y."

We define the s-dimensional monomial representations (here $\gamma = e^{j2\pi/m}$):

$$T^i(a) = \operatorname{diag}(\gamma^i, \gamma^{ir}, \dots, \gamma^{ir^{s-1}}),$$

$$T^i(b) = \begin{bmatrix} 0 & 0 & \cdots & 0 & \gamma^{it} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \qquad i = 1, \dots, m.$$

Following Curtis and Reiner (1966), we have

(i) Any irreducible representation of G is equivalent to some component in the direct sum decomposition of one of the T^i.

(ii) In fact, every irreducible representation of G is one dimensional or is equivalent to one of the T^i if and only if

$$r^j i = i \bmod m \Rightarrow ri = i \bmod m \qquad \forall i = 1, \dots, m \text{ and } j = 1, \dots, s-1 \tag{56}$$

In fact, if we let $d = \gcd(r-1, m)$, then there are precisely sd inequivalent one dimensional representations and $(m-d)/s$ inequivalent irreducible T^i.

(iii) The condition (56) is satisfied if s is a prime.

The details of the description of the set of irreducible G-representations is in general quite complex (see Curtis and Reiner (1966)). For our purposes, we need only take advantage of fact (i) above, which tells us that all of the G-transform matrices can be readily determined if we know the monomial transform matrices

$$C^i(\phi) = \frac{1}{m} \sum_{g \in G} \phi_g [T^i(g^{-1})]'.$$

For any $\phi \in C[G]$, we write

$$\phi = \sum_{i=0}^{m-1} \sum_{l=0}^{s-1} \phi_{il} a^i b^l.$$

Thus we have s distinct restrictions of ϕ to the cosets of H. Since $H \simeq Z_m$, we regard these as elements of $C[Z_m]$ and compute the Z_m transforms

$$\eta_{k,i}(\phi) = \frac{1}{m} \sum_{l=0}^{m-1} \phi_{l,i} \gamma^{-kl}, \qquad k = 0, 1, \dots, m-1, \quad i = 0, 1, \dots, s-1. \tag{57}$$

Then (dropping the dependence on ϕ for simplicity)

$$C^i(\phi) = \begin{bmatrix} \eta_{i,0} & \gamma^{-it}\eta_{i,s-1} & \cdots & \gamma^{-it}\eta_{i,1} \\ \eta_{ir,1} & \eta_{ir,0} & \cdots & \gamma^{-it}\eta_{ir,2} \\ \vdots & \vdots & & \vdots \\ \eta_{ir^{s-1},s-1} & \eta_{ir^{s-1},s-2} & \cdots & \eta_{ir^{s-1},0} \end{bmatrix} \tag{58}$$

(every term above the main diagonal has a factor of γ^{-it}). Examining (57), (58), we see that we can readily derive a fast metacyclic transform. We perform s FFT's of length m to compute (57) and then calculate (58) or directly the G-transform matrices, whose elements are linear combinations of the elements of the C^i. The amount of calculation involved is $0(n \log m)$.

We also have the duality result we have seen in the earlier cases. That is for ϕ, $\psi \in C[G]$, we have that the direct computation of $\phi * \psi$ requires $n^2 = s^2m^2$ multiplications. In the transform domain, we must perform the "pointwise" multiplication of the irreducible G-transform matrices, which involves far less computation. For example, if (56) holds, then, allowing for s^3 multiplications to compute the product of two s by s matrices, we require $sd + (m - d)\, s^2$ multiplications, which can be substantially less. If (56) does not hold, there are some irreducible representations of dimension between 1 and s, and, in fact, the above bound can be made smaller. Also, we can tighten this bound if we use efficient matrix multiplication techniques (Hopcroft and Kerr, 1968; Strassen, 1969; Borodin and Munro, 1975), as we did in the dihedral example. We also note that one can compute the transform version directly from pointwise products of "twisted" versions of the Z_n transforms. For example,

$$\eta_{i,0}(\phi * \psi) = m\eta_{i,0}(\phi)\, \eta_{i,0}(\psi) + m\gamma^{-it} \sum_{j=1}^{s-1} \eta_{i,s-j}(\phi)\, \eta_{ir^j,j}(\psi).$$

If we examine the pointwise product of ϕ and ψ, we see that this requires sm multiplications in the group algebra domain. In the transform domain, we can consider each of the H-cosets separately, thus obtaining

$$\eta_{k,i}(\phi\psi) = \sum_{l=0}^{m-1} \eta_{l,i}(\phi)\, \eta_{k-l,i}(\phi), \qquad i = 0,\ldots, s-1,$$

which is a set of s length m cyclic convolutions, which requires sm^2 multiplications. Thus again we see the pointwise-convolution duality, and, with the aid of a fast transform technique, we are able to exploit this duality to obtain a fast metacyclic filtering algorithm.

7. The Prediction and Smoothing Problems

We again consider the model (2), (3), and first let us solve the prediction problem—i.e., the computation of $\rho(k \mid l)$ for $k > l$. As we saw in Section 3, the filtering algorithm can be used to compute $\rho(l \mid l)$, which we take as our initial condition for the prediction problem. The independence assumption on the noise sequences then allows us to calculate (here we assume $U = X$, $b =$ identity)

$$\rho(k+1 \mid l) = \eta(k) * a[\rho(k \mid l)], \qquad k \geqslant l.$$

We see that the computations involved here are identical to those for the diffusion update in the filtering algorithm.

To solve the smoothing problem—i.e., the recursive computation of $\rho(k \mid m)$ as a function of $m \geqslant k$, with initial condition $\rho(k \mid k)$—we must introduce some new maps. First define M_1, M_2: $C[X \times X] \to C[X]$.

$$M_1 \left[\sum_{g,h \in X} \alpha_{gh} \cdot (g, h) \right] = \sum_{g \in X} \left(\sum_{h \in X} \alpha_{gh} \right) \cdot g,$$

$$M_2 \left[\sum_{g,h \in X} \alpha_{gh} \cdot (g, h) \right] = \sum_{h \in X} \left(\sum_{g \in X} \alpha_{gh} \right) \cdot h.$$

Note that M_1 is the extension of the homomorphism $(g, h) \to g$, while M_2 is the extension of $(g, h) \to h$. For any $t \in X$, we also define p_t: $C[X \times X] \to C[X]$

$$p_t \left[\sum_{g,h \in X} \alpha_{gh} \cdot (g, h) \right] = \sum_{g \in X} \alpha_{gt} \cdot g.$$

This is simply the restriction of a function to the coset $X \times \{t\}$. Finally, we define a pointwise product "$\circ$" between elements of $C[X]$ and $C[X \times X]$

$$\left(\sum_{t \in X} \beta_t \cdot t \right) \circ \left(\sum_{g,h \in X} \alpha_{gh} \cdot (g, h) \right) = \sum_{g,h \in X} \beta_g \alpha_{gh} \cdot (g, h).$$

Proposition 2. The solution to the smoothing problem is

$$\lambda(m) = \xi(m)\, y(m), \tag{59}$$

$$\psi(m, k) = \hat{c}[\lambda(m)] \circ \phi(m, k), \tag{60}$$

$$\gamma(k \mid m) = \rho(k \mid m-1)\, M_2[\psi(m, k)], \tag{61}$$

$$N(k \mid m) = \sum_{g \in X} \gamma(k \mid m)_g, \tag{62}$$

$$\rho(k \mid m) = \gamma(k \mid m)/N(k \mid m), \tag{63}$$

where

$$\phi(m, k)_{(g,h)} = \Pr[x(m) = g \mid x(k) = h, y(k+1),..., y(m-1)] \tag{64}$$

The proof of this result is similar to the proof of Proposition 1, and we refer the reader to Willsky (1976) for a proof. Let us briefly examine the computational issues involved. The measurement permutation (59) and the pullback $\hat{c}$ are the same as before, and the same comments hold. The quantity $\phi(m, k)$ is computed "column by column"—i.e., we compute $p_g[\phi(m, k)]$ for each $g \in X$. But $p_g[\phi(m, k)]$ is just the distribution for $x(m)$ given we *know* that $x(k) = g$ and given the observations $y(k+1),..., y(m+1)$. This is just a filtering problem with initial condition $\rho(k \mid k) = g$ (an impulse), and the computations involved in such problems have been discussed in earlier sections. The pointwise product $\hat{c}[\lambda] \circ \phi$ involves many of the issues involved in usual pointwise products, and we leave the details of this to the reader. Also, since M_2 is an extension, the calculation of $M_2(\psi)$ can be understood in terms of the concepts introduced in earlier sections. Finally the pointwise product $\rho M(\psi)$ and the normalization (62), (63) are identical, as far as computational complexity goes, to the corresponding operations in the filtering algorithm. Detailed analyses can, of course, be worked out in specific cases.

8. Further Transform Calculations for Z_n-Random Processes

In this section we illustrate some other issues that can be studied using transform ideas. We do this in the simplest setting (i.e., Z_n) but the concepts carry over to the general finite group case.

A. *Convolution and Translation-Stable Densities*

Definition. A class $\mathscr{F}$ of probability distributions on Z_n is *convolution stable* if we have $p_1 * p_2 \in \mathscr{F}$ whenever $p_1, p_2 \in \mathscr{F}$. A convolution stable class is *translation stable* if $f * \delta_j \in \mathscr{F}$ for $j \in Z_n$, whenever $f \in \mathscr{F}$.

Suppose that a particular convolution stable class is parametrized by a single, nonnegative real variable α. Let us denote the parametrization by a subscript—i.e., $p_\alpha \in \mathscr{F}$ $\alpha > 0$. Let us also define

$$C^i(p_\alpha) = D^i(\alpha)/n,$$

and let us assume that the parametrization is such that

$$p_{\alpha_1} * p_{\alpha_2} = p_{\alpha_1+\alpha_2} \qquad \text{or} \qquad D^i(\alpha_1)\, D^i(\alpha_2) = D^i(\alpha_1 + \alpha_2). \tag{65}$$

If we assume a continuous dependence on the parameter α, we have

$$D^i(\alpha) = e^{\alpha\beta(i)}$$

for some function β such that the resulting inverse transform is a valid probability distribution. A valid example is

$$\beta(k) = -\binom{n}{k} = -\frac{n!}{k!(n-k)!}. \tag{66}$$

In this case the distribution has zero mode and is symmetric about the mode.

Given a convolution stable class $\mathscr{F}$ we can construct a translation stable class by adjoining to $\mathscr{F}$ all distributions $f * \delta_j$ with $f \in \mathscr{F}$, $j \in Z_n$. Applying this construction to the class defined by (66), we obtain the class

$$S(j;\eta,\gamma) = \frac{1}{n} + \frac{1}{n}\sum_{k=1}^{n-1} \exp\left[-\alpha\binom{n}{k}\right]\cos\frac{2\pi k(j-\eta)}{n},$$

where $\alpha \geqslant 0$, $\eta \in Z_n$. $S(j;\eta,\gamma)$ has η as its mode and is symmetric about it.

We note that one can also consider classes $\mathscr{F}$ that are stable under extensions or pullbacks. The use of transforms should be as useful for the analysis of such classes as well. Note that if one has a stable class, the filtering algorithm may be simplified—i.e., we need only keep track of the evolution of the parameters that specify the distribution. Issues such as these await future investigation.

B. *Optimal Estimation*

Let x be a Z_n-random variable with distribution p. Note that p being real implies (here "‾" denotes complex conjugate)

$$C^i(p) = \overline{C^{n-i}(p)}. \tag{67}$$

Let ϕ be a real-valued function on Z_n given by

$$\phi(l) = \sum_{k=0}^{n-1} d_k\gamma^{-kl}.$$

Suppose we wish to choose the estimate $\hat{x} \in Z_n$ that minimizes

$$E[\phi(x-\hat{x})] = n\sum_{k=0}^{n-1} C^k(p)\, d_k\gamma^{k\hat{x}}. \tag{68}$$

In general, the minimization of (68) requires a search over all $\hat{x} \in Z_n$; however, in some cases this process can be simplified. For example, consider the cost criterion obtained by choosing $\phi(j) = 1 - \cos(2\pi j/n)$. The Fourier decomposition of ϕ yields

$$d_0 = 1, \qquad d_1 = d_{n-1} = -1/2,$$

with all other $d_k = 0$. Using (67), we have that (68) reduces to

$$\frac{1}{n} E[\phi(x - \hat{x})] = 1 - \left[\operatorname{Re}(c_1) \cos \frac{2\pi\hat{x}}{n} - \operatorname{Im}(c_1) \sin \frac{2\pi\hat{x}}{n}\right].$$

A straightforward calculation yields the optimal estimate

$$\hat{x} = \left[\frac{n}{2\pi} \tan^{-1} \left\{- \frac{\operatorname{Im}(c_1)}{\operatorname{Re}(c_1)}\right\}\right]_n, \tag{69}$$

where $[\alpha]_n$ = the integer closest to α (modulo n). We note that this estimation criterion yields the mode of p if p is unimodal and symmetric about its mode, and it allows us to compute an estimate directly as a function of the transform of the distribution p.

C. *Nonhomomorphic Observations*

We want to show that, although we lose the nice structure of the pull-back mapping, the pointwise product-convolution duality is preserved even if one has nonhomomorphic measurements. To do this it is enough to consider a static problem. Let x be a Z_n-random variable with distribution ρ, and suppose we observe y. All we need to know about this observation is the distribution (if y is discrete) or the density (if y is continuous) $p(y \mid x = i)$. Using Bayes' rule, we then have

$$\eta(i) = \Pr(x = i \mid y) = \frac{p(y \mid x = i)\, \rho(i)}{\sum_{l=0}^{n-1} p(y \mid x = l)\, \rho(l)}. \tag{70}$$

Regarding $p(y \mid x = i)$ as a function of i, we see that the numerator of (70) consists of a pointwise multiplication of two elements of $C[Z_n]$. Here the observed value y simply tells us which function $p(y \mid x = i)$ we want.

We can also write a dual algorithm. Write

$$p(y \mid x = l) = \sum_{k=0}^{n-1} h_k(y) \gamma^{kl},$$

where the h_k are functions of y (see Willsky (1976) for an example). It is then clear that

$$C^i(\eta) = B^i / nB^0,$$

$$B^i = \sum_{r=0}^{n-1} h_r(y)\, C^{i-r}(\rho),$$

and we have the desired convolution result.

D. *A Continuous-Time Estimation Problem on Z_n*

In this section we illustrate how the transform framework can be used to study continuous-time Z_n estimation problems. We do this by considering a specific phase tracking problem. The formulation presented here is similar to that used in phase-shift-keying (PSK) communication problems (Stiffler, 1971; Lindsey, 1966).

Let $x(t)$ be a continuous-time jump process on Z_n and let $p_x(t)$ denote its distribution. We assume that p_x satisfies the differential equation

$$\dot{p}_x(t) = \alpha(t) * p_x(t). \tag{71}$$

We note that such a process is the continuous-time analog of the process in (32) $a = 1$ (for more on this, see Willsky (1976)).

We now consider a continuous-time observation process

$$dz(t) = \left[\sin\left(\frac{2\pi x(t)}{n}\right)\right] dt + r^{1/2}(t)\, dv(t),$$

where $r(t) > 0$ and v is a Brownian motion process independent of x. We wish to compute the conditional distribution $p(t)$ of $x(t)$ given $z(s)$, $s \leqslant t$. Using a result from Wonham (1965), we find that p satisfies

$$dp(t)_l = [\alpha(t) * p(t)]_l\, dt + \frac{[\sin(2\pi l/n) - \hat{h}(t)]}{r(t)} [dz(t) - \hat{h}(t)\, dt]\, p(t)_l\,, \tag{72}$$

$$\hat{h}(t) = E\left[\sin \frac{2\pi x(t)}{n} \,\middle|\, z(s),\, 0 \leqslant s \leqslant t\right]. \tag{73}$$

Thus, the computation of $p(t)$ involves the cyclic convolution (71), the computation of $\hat{h}$ in (73), and the pointwise product to calculate the second term on the right-hand side of (72).

The computations in the transform domain are decidely simpler. Let

$$q_k(t) = \frac{1}{n} \sum_{l=0}^{n-1} \alpha_l(t)\gamma^{-kl}, \qquad c_k(t) = \frac{1}{n} \sum_{l=0}^{n-1} p(t)_l\gamma^{-kl}.$$

Then

$$\hat{h}(t) = -n\,\mathrm{Im}[c_1(t)],$$

$$dc_k(t) = nq_k(t)\, c_k(t)\, dt,$$

$$+ \left[\frac{dz(t) + n\,\mathrm{Im}[c_1(t)]}{r(t)}\right]\left[\frac{c_{k-1}(t) - c_{k+1}(t)}{2j} + nc_k(t)\,\mathrm{Im}[c_1(t)]\right]. \tag{74}$$

From (67) we see that we need only compute c_k for $k = 1,..., \lceil(n - 1)/2\rceil$. Also, the right-hand side of (74) involves very few multiplications.

We note that there is no difficulty in considering an observation with a known carrier frequency. In this case our received signal is

$$\begin{aligned} dz(t) &= \sin\left(w_c t + \frac{2\pi x(t)}{n}\right) dt + r^{1/2}(t)\, dv(t) \\ &= \sin w_c t \cos \frac{2\pi x(t)}{n}\, dt + \cos w_c t \sin \frac{2\pi x(t)}{n}\, dt + r^{1/2}(t)\, dv(t), \end{aligned}$$

and our filter becomes time-varying, but the filter structure remains essentially unchanged.

Thus we see that in this example we have a far simpler implementation of the filtering algorithm in the transform domain. The convolution (71) is avoided, $\hat{h}$ is calculated easily in terms of the transform, the measurement update convolution (essentially the second term on the right-hand side of (74) is extremely sparse, as only c_1, c_{k-1}, c_{k+1} couple into c_k, and, if we use the cost criterion, $\phi(j) = 1 - \cos(2\pi j/n)$, the optimal estimate can be calculated easily in terms of c_1 (see (69)).

9. Conclusions and Discussion

In this paper we have studied a class of estimation problems on finite groups. By viewing probability distributions as elements of a group algebra, and, by taking the transforms of such elements, we were able to uncover the underlying structure of the filtering problem. We have illustrated this structure by means of several examples which display the duality between the two proposed filtering algorithms. Also by utilizing fast Fourier transform techniques and a generalization of the FFT to metacyclic groups, we have been able to point out an efficient realization of the filtering solution. We note that Depeyrot (1968, 1971) has considered several of these issues (although none of the filtering aspects and hence none of the duality issues), but he has limited himself to the so-called *character transforms*, which utilize only the irreducible group characters (Curtis and Reiner, 1966). Since *all* characters are constant on conjugacy classes, a given character transform corresponds to many elements of the algebra unless the group is abelian.

We have also briefly explored a number of other problems that can be analyzed within our framework. These include prediction and smoothing, characterization of stable distribution classes, optimal estimation, nonhomomorphic measurements, and continuous-time problems. In the last of these we considered a problem often encountered in synchronous communication and have proposed what we feel is an efficient implementation.

The potential saving in computational burden for problems that can be put into our framework may have an impact on some finite state Markov process control problems, such as those considered in Astrom (1965). In these problems the control is a function of the conditional distribution of the state, which must be computed on-line. If the particular process could be viewed as evolving on a finite group the problem could be cast into our framework, yielding computational benefits. Also, Sandell (1974) considers a limited memory version of the control problem, and derives an off-line procedure for determining the optimal control law (here a function of the finite-state memory). Again, for those problems that can be placed into the finite group setting we may be able to achieve great computational savings in the off-line calculations as well as for those that must be performed on-line.

Finally, let us comment on several possible extensions of the ideas presented in this paper. First of all, we note that there are quite likely to be far larger classes of groups for which fast transforms exist. A likely place to start is to investigate other group extensions beyond the metacyclic example considered in this paper. In addition, all of the algorithms considered in this paper have been defined over the complex numbers. This was done to guarantee that $C[X]$ was semisimple and that C was a splitting field for X (i.e., no irreducible C-representation is reducible over any extension field of C—see Curtis and Reiner (1966)). Quite often one can use other fields (e.g., finite fields) over which the implementation of the fast transform algorithms are far simpler. For example, this is true for the so-called "number-theoretic transforms," which are simply transforms in $K[Z_n]$ for certain finite fields K (see Nicholson, 1971; Agarwal and Burrus, 1975). Further general results on the computational complexity of convolution in $K[G]$ for K a field and G a finite group are reported in Loui (1976).

Extensions of our work are possible along several other lines. First of all, the directions explored in Sections 7 and 8 remain as areas for further work. For example, the study of parametrizable stable distribution classes may lead to efficient filtering algorithms, in which we need only track the parameters. Finally, we note that as discussed in Depeyrot (1968), Paz (1971), and Willsky (1973), by considering the extension of an arbitrary POFSMP to a process evolving on a finite semigroup (essentially the Myhill semigroup of a finite state automaton that realizes the given POFSMP), we obtain equations similar to (2), (3) but over a finite semigroup (and perhaps with a nonhomomorphic output). One is then led naturally to the development of dual filtering algorithms over semigroups, and this requires the study of semigroup algebras and representations of finite semigroups. Intuitively, the complexity of semigroup algebra convolution should be less than for a "comparable" group, since the semigroup product will "collapse" several terms together, thus trading off multiplications for additions. This vague statement has been illustrated in an example in Willsky (1973) and is corroborated for finite cyclic semigroups in Loui (1976). Results beyond these initial ones await further study. This appears to be a useful direction for further work, since in principle *any* POFSMP can be cast in this framework, and the computational savings to be obtained from further research would be extremely useful in combatting the combinatorics problems that run rampant in estimation and control problems over finite state sets.

Appendix

A.1. *Calculation of the Transform of Pointwise Products*

Let X be a finite group with cardinality n, and let $T^1,..., T^s$ be a complete set of inequivalent orreducible representations, with $\dim T^1 = z_i$. Let ρ, $\psi \in C[X]$. We wish to compute $C^i(\rho\psi)$. Examining the equation

$$\rho_g\psi_g = \left[\sum_{i=1}^{s} \sum_{j,k=1}^{z_i} C^i_{jk}(\rho)\, t^i_{jk}(g)\right]\left[\sum_{l=1}^{s} \sum_{p,q=1}^{z_l} C^l_{pq}(\psi)\, t^l_{pq}(g)\right], \tag{75}$$

we see that we are faced with products of the form

$$t^i_{jk}(g)\, t^l_{pq}(g). \tag{76}$$

Such functions can be obtained as matrix elements of the tensor product representation $T^i \otimes T^l$, where $T^i(g) \otimes T^l(g)$ is the Kronecker product of $T^i(g)$ and $T^l(g)$. For example, the term in (76) is the $((j-1)\, z_l + p,\, (k-1)\, z_l + q)$ element of $T^i \otimes T^l$. From this and from the forms of (13) and (75), we can compute the $C^r(\rho\psi)$. Define the *characteristic matrices*

$$L^{(i,\alpha,\gamma)}_{\eta\xi} = \frac{z_\gamma}{n} \sum_{g\in X} T^i(g) \otimes T^\alpha(g)\, t^\gamma_{\xi\eta}(g^{-1}). \tag{77}$$

Then

$$C^r_{\eta\xi}(\rho\psi) = \sum_{\substack{i,j,k\\ \alpha,\beta,\gamma}} C^i_{jk}(\rho)\, C^\alpha_{\beta\gamma}(\psi)[L^{(i,\alpha,\gamma)}_{\eta\xi}]_{(j-1)z_\alpha+\beta,\,(k-1)z_\alpha+\gamma}. \tag{78}$$

The characteristic matrices can be found as follows. The representation $T^i \otimes T^\alpha$ is equivalent to a direct sum of irreducible representations—i.e., there exists an invertible matrix $P_{i\alpha}$ such that

$$P^{-1}_{i\alpha}(T^i \otimes T^\alpha)P_{i\alpha} = \mathrm{diag}(T^{k(1)},..., T^{k(r)}), \tag{79}$$

where the integers r, $k(1),..., k(r)$ depend on i and α. Using (79), the definition of $L^{(i,\alpha,\gamma)}_{\eta\xi}$, and the orthogonality relation (11), we see that

$$P^{-1}_{i\alpha}L^{(i,\alpha,\gamma)}_{\eta\xi}P_{i\alpha} = (E^\gamma_{\eta\xi}\delta_{\gamma,k(1)},\ldots,E^\gamma_{\eta\xi}\delta_{\gamma,k(r)}). \tag{80}$$

We note that in general most of the $L^{(i,\alpha,\gamma)}$ are zero. Also, there exists a direct method for determining r, $k(1),\ldots, k(r)$ by examination of the group characters. We refer the reader to Willsky (1976) for the discussion of this method and for the calculation of the characteristic matrices and of $C^i(\phi\psi)$ for $X = D_n$, the dihedral group on n letters.

A.2. *Calculation of Extensions and Pullbacks*

The general calculation of the functions f_i and $\hat{f}_i$ defined in (17), (18) is given in Willsky (1976). We content ourselves here with the derivation of the equations needed for our filtering algorithm.

As discussed in Section 3, the only extension used in the filtering algorithm is a: $C[X] \rightarrow C[X]$. Note first that if S is an irreducible representation of $a(X)$, then $S \circ a$ is an irreducible representation of X. Thus, let $S^1,\ldots, S^\nu$ be a complete set of irreducible representations of $a(X)$, chosen and ordered so that

$$T^i = P_i(S^i \circ a)P_i^{-1}, \qquad i = 1,\ldots, \nu. \tag{81}$$

Now, let $\rho \in C[X]$. Then, a straightforward calculation yields the transform matrices $D^i(\alpha(\rho))$ where $a(\rho)$ is regarded as an element of $C[a(X)]$:

$$D^i(a(\rho)) = tP_i'C^i(\rho)(P_i')^{-1}, \qquad i = 1,\ldots, \nu, \tag{82}$$

where $|X| = n$, $|a(X)| = n/t$.

The problem now is to determine the transform matrices $C^i(a(\rho))$, of $a(\rho)$ regarded as an element of $C[X]$. We first note that each of the T^i, when restricted to $a(X)$, is not necessarily irreducible, but we can write

$$T^i = R_i \operatorname{diag}(S^{l(1)},\ldots, S^{l(\mu)})R_i^{-1}, \tag{83}$$

where μ, $l(1),\ldots, l(\mu)$ depend on i. Then combining (82), (83), another simple calculation yields

$$\begin{aligned} a_i(C(\rho)) &= C^i(a(\rho)) \\ &= z_i(R_i')^{-1} \operatorname{diag}\Big(\frac{1}{z_{l(1)}} P'_{l(1)}C^{l(1)}(\rho)(P'_{l(1)})^{-1},\ldots, \frac{1}{z_{l(\mu)}} \\ &\quad \times P'_{l(\mu)}C^{l(\mu)}(\rho)(P'_{l(\mu)})^{-1}\Big)R_i'. \end{aligned} \tag{84}$$

Thus, we see that the required calculation (84) consists of similarity transformations, which, by judicious choice of the basic representations (the S's and T's) can often be made quite simple. In fact, if the T^i and S^i are monomial, then the P_i will be extremely sparse and will usually contain many entries of ± 1. The same will be true of the R_i, although they may be somewhat more complex if $\mu > 1$. Note however, that if a is an isomorphism, then the set $S^i \circ a$ is also a complete set of irreducible representations of X, $\mu = 1$ for all i, and (84) becomes

$$a_i(C(\rho)) = (R_i')^{-1} P_{l(1)}' C^{l(1)}(\rho)(P_{l(1)}')^{-1} R_i' . \tag{85}$$

That is, the a_i simply permute the $C^i(\rho)$ except for a similarity transformation, which, as our examples indicate, if often the identity.

We now examine the pullback map $\hat{c}$ of the homomorphism $c\colon X \to Y$, which is needed in the calculations in equations (20) and (25). The calculation of $\hat{c}^i(C(\lambda))$ for $\lambda \in C[Y]$ consists of two steps, the first of which is necessary only if c is not surjective. Let $U^1,\ldots, U^r$ be a complete set of irreducible representations of Y ($\dim U^i = v_i$), let $V^1,\ldots, V^\nu$ be an analogous set for $c(X)$ ($\dim V^i = w_i$), let $\lambda_R \in C[c(X)]$ denote the restriction of λ to $c(X)$, and let $|\, c(X)| = d$. We first wish to compute the $c(X)$-transform matrices $D^i(\lambda_R)$ when we are given the Y-transform $C^j(\lambda)$. As before, the U^i need not be irreducible when restricted to $c(X)$. Hence, on $c(X)$ we have

$$U^l = M_l \operatorname{diag}(V^{\epsilon(1)},\ldots, V^{\epsilon(q)}) M_l^{-1}, \tag{86}$$

where q, $\epsilon(1),\ldots, \epsilon(q)$ depend on l.

We now must compute the $c(X)$-transform of the elements of the U^l restricted to $c(X)$. Thus, we define the (precomputable) *subgroup matrices*

$$L_{\alpha\beta}^{q,i} = \frac{w_i}{d} \sum_{h \in c(X)} U^q(h)\, S_{\beta\alpha}^i(h^{-1}). \tag{87}$$

Then, using (86) and (87), we compute

$$L_{\alpha\beta}^{q,i} = M_i \operatorname{diag}(E_{\alpha\beta}^i \delta_{i,\epsilon(1)}, \ldots, E_{\alpha\beta}^i \delta_{i,\epsilon(q)}) M_i^{-1}. \tag{88}$$

A direct calculation (Willsky, 1976) then yields

$$D_{\alpha\beta}^i(\lambda_R) = \sum_{l=1}^{r} \sum_{j,k=1}^{v_l} C_{jk}^l(\lambda)[L_{\alpha\beta}^{l,i}]_{jk} . \tag{89}$$

Note that if $c(X) = Y$, $D^i = C^i$ $\forall i$.

Having the D^i, we now want to compute the X-transform $\hat{c}_i(C(\lambda)) = C^i(\hat{c}(\lambda)) = C^i(\hat{c}(\lambda_R))$. For each T^i, define the function on $c(X)$

$$\tilde{T}^i(g) = \frac{d}{n} \sum_{h \in c^{-1}(g)} T^i(h), \qquad g \in c(X). \tag{90}$$

Note that $\tilde{T}^i$ may equal 0, but if it does not, it is a (possibly reducible) representation of $c(X)$. In this case, we can write

$$\tilde{T}^i = Q_i \operatorname{diag}(V^{\sigma(1)},..., V^{\sigma(\alpha)})Q_i^{-1}, \tag{91}$$

where α, $\sigma(1),..., \sigma(\alpha)$ depend on i. Using (9), (90), and (91), a straightforward computation yields

$$\hat{c}_i(C(\lambda)) = z_i(Q_i')^{-1} \operatorname{diag}\left(\frac{1}{w_{\sigma(1)}} D^{\sigma(1)}(\lambda_R),..., \frac{1}{w_{\sigma(\alpha)}} D^{\sigma(\alpha)}(\lambda_R)\right) Q_i' \tag{92}$$

if $\tilde{T}_i \neq 0$, and $\hat{c}_i(C(\lambda)) = 0$ if $\tilde{T}_i = 0$.

Thus, the required on-line calculations are given in (89) and (92). Again, the calculation (89) is only necessary if c is not surjective. In any event, we note that (89) requires computing only certain linear combinations of the $C_{jk}^i(\lambda)$, and in general the subgroup matrices will be quite sparse. In addition, many of the U^i, when restricted to $c(X)$, may still be irreducible (as they all are in the abelian case), and in this case the calculations become even simpler. As for the calculation of (92), again this only involves similarity transformations, and the sparseness of the Q_i, combined with the possible irreducibility of the $\tilde{T}_i$, often make the calculation of $\hat{c}_i$ quite simple.

Finally, we note that one can directly determine the integers μ, $l(1),..., l(\mu)$ in (83) q, $\epsilon(1),..., \epsilon(q)$ in (86) and α, $\sigma(1),..., \sigma(\alpha)$ in (91) with the aid of group characters for the various representations. We refer the reader to Willsky (1976) for details.

A.3. *Proof of Proposition* 1

Consider the system (2), (3) with the independence assumptions and notation introduced in Section 2. We first prove the validity of the diffusion update equation (19). Thus, suppose we have $\rho(k \mid k)$. Then, using the independence assumption on $u(k)$, we have

$$\begin{aligned}
\rho(k+1 \mid k)_g &= \Pr(x(k+1) = g \mid y(1),..., y(k)) \\
&= \sum_{h \in X} \Pr(b[u(k)] = h) \Pr(a[x(k)] = h^{-1}g \mid y(1),..., y(k)) \\
&= \sum_{h \in X} \left\{ \sum_{\mu \in b^{-1}(h)} \Pr(u(k) = \mu) \right\} \left\{ \sum_{\nu \in a^{-1}(h^{-1}g)} \Pr(x(k) = \nu \mid y(1),..., y(k)) \right\} \\
&= \sum_{h \in X} b[\eta(k)]_h \, a[\rho(k \mid k)]_{h^{-1}g} = \{b[\eta(k)] * a[\rho(k \mid k)]\}_g .
\end{aligned}$$

Now we assume we have $\rho(k \mid k-1)$. Then, using Bayes' rule and the independence of the $v(k)$, we have

$$\rho(k \mid k)g = \Pr[x(k) = g \mid y(1),\ldots, y(k)]$$

$$= \frac{\Pr[y(k) \mid x(k) = g, y(1),\ldots, y(k-1)] \Pr[x(k) = g \mid y(0),\ldots, y(k-1)]}{\sum_{h\in X} \Pr[y(k) \mid x(k) = h, y(1),\ldots, y(k-1)] \Pr[x(k) = h \mid y(0),\ldots, y(k-1)]}$$

$$= \frac{\Pr[y(k) \mid x(k) = g]\, \rho(k \mid k-1)_g}{\sum_{h\in X} \Pr[y(k) \mid x(k) = h]\, \rho(k \mid k-1)_h}\,.$$

Noting the definitions of $\lambda(k)$ and $\gamma(k \mid k)$ in (20), (21), we see that our update equations will be shown to be valid once we have shown that

$$\hat{c}[\xi(k)\, y(k)]_g = \Pr[y(k) \mid x(k) = g]. \tag{93}$$

Rewriting the right-hand side of (93), we obtain

$$\begin{aligned} \Pr[y(k) \mid x(k) = g] &= \Pr[v(k) = y(k)\, c(g)^{-1}] \\ &= \Pr[v^{-1}(k) = c(g)\, y(k)^{-1}] = \xi(k)_{c(g)y(k)^{-1}}\,. \end{aligned}$$

The definition of the pullback map yields

$$\begin{aligned} \hat{c}[\xi(k) * y(k)] &= \hat{c}\left\{\sum_{h\in y} \xi(k)_h \cdot (hy(k))]\right\} \\ &= \sum_{h\in y} \xi(k)_h \left\{ \sum_{t\in c^{-1}[hy(k)]} t \right\}. \end{aligned} \tag{94}$$

We wish to compute $\hat{c}[\xi(k)\, y(k)]_g$—i.e., the g-component of $\hat{c}[\xi(k)\, y(k)]$. Examining (94), we see that

$$g \in c^{-1}[hy(k)] \Leftrightarrow h = c(g)\, y(k)^{-1}.$$

Thus

$$\hat{c}[\xi(k)\, y(k)]_g = \xi(k)_{c(g)y(k)^{-1}}\,,$$

and the result is proved.

RECEIVED: May 13, 1977; REVISED: October 21, 1977

REFERENCES

AGARWAL, R. C., AND BURRUS, C. S. (1975), Number theoretic transforms to implement fast digital convolution, *Proc. IEEE* **63**, 550–560.

ASTROM, K. J. (1965), Optimal control of Markov processes with incomplete state information, *J. Math. Anal. Appl.* **10**, 174–205.

Bharucha-Reid, A. T. (1960), "Elements of the Theory of Markov Processes and Their Applications," McGraw–Hill, New York.

Borodin, A., and Munro, I. (1975), "The Computational Complexity of Algebraic and Numeric Problems," American Elsevier, New York.

Brockett, R. W., and Willsky, A. S. (1972), Finite group homomorphic sequential systems, *IEEE Trans. Automatic Control* **AC-17**, 483–490.

Chizeck, H. (1976), "Inverse and Coding Applications for Systems Evolving in Finite groups," S.M. Thesis, Dept. of System Engineering, Case Western Reserve Univ., Cleveland, Ohio.

Curtis, C. W., and Reiner, I. (1966), "Representation Theory of Finite Groups and Associative Algebras," Interscience, New York.

Depeyrot, M. (1968), "Operand Investigation of Stochastic Systems," Ph.D. Dissertation, Stanford University.

Depeyrot, M., Marmorat, J. P., and Mondelli, J. (1971), An automaton theoretic approach to the F.F.T., *in* "Proceedings of a Symposium on Computers and Automata," p. 359, Polytechnic Press, New York.

Depeyrot, M. (1974), Approache barycentrique par transformation de Fourier d'une classe de problèmes de placement et de transport, *C. R. Acad. Sci. Paris* **279**.

Fliess (1972), "Sur certaines familles de séries formelles," Thèse de Doctorat d'État, Université Paris VII.

Fliess (1976), Un outil algébrique: les séries formelles non commutatives, *in* "Mathematical Systems Theory" (G. Marchesini and S. K. Mitter, Eds.), Lecture Notes in Econ. and Math. Systems, Springer-Verlag, New York.

Good, I. J. (1958), The interaction algorithm and practical Fourier analysis, *J. Roy. Statist. Soc. B* **20**, 361–372.

Grenander, U. (1963), "Probabilities on Algebraic Structures," Wiley, New York.

Heading, J. (1958), "Matrix Theory for Physicists," Longmans, Green, London.

Hopcroft, J., and Kerr, L. (1968), On minimizing the number of multiplications necessary for matrix multiplication, *SIAM J. Appl. Math.* **20**, 20–36.

Jazwinski, A. H. (1970), "Stochastic Processes and Filtering Theory," Academic Press, New York.

Lindsey, W. C. (1966), Phase-shift-keyed signal detection with noisy reference signals, *IEEE Trans. Aero. and Electron. Syst.* **AES-2**, 393–401.

Loui, M. C. (1976), "Efficient Multiplication in Semisimple Algebras," S.M. Thesis, Dept. of Elec. Eng. and Comp. Sci., M.I.T., Cambridge, Mass.

Nicholson, P. J. (1971), Algebraic theory of finite Fourier transforms, *J. Comput. System Sci.* **5**, 524–547.

Oppenheim, A. V., and Schafer, R. W. (1975), "Digital Signal Processing," Prentice–Hall, Englewood Cliffs, N.J.

Paz, A. (1971), "Introduction to Probabilistic Automata," Academic Press, New York.

Rotman, J. (1965), "The Theory of Groups: An Introduction," Allyn & Bacon, Boston.

Sandell, N. R. (1974), "Control of Finite-State, Finite-Memory Stochastic Systems," Rept. ESL-R-545, M.I.T. Electronic Systems Laboratory, Cambridge, Mass.

Stiffler, J. J. (1971), "Theory of Synchronous Communication," Prentice–Hall, Englewood Cliffs, N.J.

Strassen, V. (1969), Gaussian elimination is not optimal, *Numer. Math.* **13**, 354–356.

Willsky, A. S. (1973), "Dynamical Systems Defined on Groups: Structural Properties and Estimation," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, M.I.T., Cambridge, Mass.

Willsky, A. S. (1974), A finite Fourier transform approach to estimation on cyclic groups, *in* "Proceedings of the Fifth Symposium on Nonlinear Estimation and Its Applications, San Diego, Calif."

WILLSKY, A. S. (1975), Invertibility of finite group homomorphic sequential systems, *Inform. Contr.* **27**, 126–147.

WILLSKY, A. S. (1976), "On the Algebraic Structure of Certain Partially Observable Finite-State Markov Processes," Rept. ESL-R-686, M.I.T., Electronic Systems Lab., M.I.T., Cambridge, Mass.

WONHAM, W. M. (1965), Some applications of stochastic differential equations to optimal nonlinear filtering, *SIAM J. Control* **2**, 347–369.