What does a nonabelian group sound like?

Harmonic Analysis on Finite Groups and DSP Applications

Matthew Corley and William DeMeo

University of South Carolina

Abstract

Underlying many digital signal processing (DSP) algorithms, in particular those used for digital audio filters, is the convolution operation, which is a weighted sum of translations $f(x-y)$. Most classical results of DSP are easily and elegantly derived if we define our functions on $\mathbb{Z}/n\mathbb{Z}$, the abelian group of integers modulo n. If we replace this underlying "index set" with a nonabelian group, then translation may be written $f(y^{-1}x)$, and the resulting audio filters arising from convolution naturally produce different effects that those obtained with ordinary (abelian group) convolution.

The goal of this project is to explore the idea of using the underlying finite group (i.e., the index set) as an adjustable parameter of a digital audio filter. By listening to samples produced using various nonabelian groups, we try to get a sense of the "acoustical characters" of finite groups.

1 Introduction

The *translation-invariance* of most classical signal processing transforms and filtering operations is largely responsible for their widespread use, and is crucial for efficient algorithmic implementation and interpretation of results [1].

DSP on *finite abelian groups* such as $\mathbb{Z}/n\mathbb{Z}$ is well understood and has great practical utility. Translations are defined using addition modulo n, and basic operations, including convolutions and Fourier expansions, are developed relative to these translations [2]. Recently, however, interest in the practical utility of *finite nonabelian groups* has grown significantly. Although the theoretical foundations of nonabelian groups is well established, application of the theory to DSP has yet to become common-place. A notable exception is [1], which develops theory and algorithms for indexing data with nonabelian groups, defining translations with a non-commutative group multiply operation, and performing typical DSP operations relative to these translations.

This paper describes the use of nonabelian groups for indexing one- and two-dimensional signals, and discusses some computational advantages and insights that can be gained from such an approach. A simple but instructive class of nonabelian groups is examined. When elements of such groups are used to index the data, and standard DSP operations are defined with respect to special group binary operators, more general and interesting signal transformations are possible.

1.1 Preview: Two Distinctions of Consequence

Abelian group DSP can be completely described in terms of a special class of signals called the *characters* of the group. (For $\mathbb{Z}/n\mathbb{Z}$, the characters are simply the exponentials.) Each character of an abelian group represents a one-dimensional translation-invariant subspace, and the set of all characters spans the space of signals indexed by the group; any such signal can be uniquely expanded as a linear combination over the characters.

In contrast, the characters of a nonabelian group G do not determine a basis for the space of signals indexed by G. However, a basis can be constructed by extending the characters of an abelian subgroup A of G, and then taking certain translations of these extensions. Some of the characters of A cannot be extended to characters of G, but only to proper subgroups of G. This presents some difficulties involving the underlying translation-invariant subspaces, some of which are now multidimensional. However, it also presents opportunities for alternative views of local signal domain information on these translation-invariant subspaces.

The other abelian/nonabelian distinction of primary importance concerns translations defined on the group. In the abelian group case, translations represent simple linear shifts in space or time. When nonabelian groups index the data, however, translations are no longer so narrowly defined.

1.2 Brief overview of nonabelian convolution

Since the main operation we will consider is convolution, we must think about how best to view this operation mathematically, as well as how best to represent it in the computer. In this section is some background on the mathematical aspects. In later sections we provide more details and examples.

Let $\mathbb{C}^G$ denote the set of complex valued functions defined on the group G. That is

$$\mathbb{C}^G = \{f : G \to \mathbb{C}\}.$$

(In the Tolimieri-An books, [2] [3], this set is also denoted by $\mathcal{L}(G)$.) If the group has $|G| = n$ elements, say, $G = \{x_0, x_1, \dots, x_{n-1}\}$, then each function $f \in \mathbb{C}^G$ can be represented as a length-n vector in $\mathbb{C}^n$—namely, the vector of its values on G:

$$\mathbf{f} = [f(x_0), f(x_1), \dots, f(x_{n-1})].$$

Given two functions f and g in $\mathbb{C}^G$, the *convolution of f and g*, denoted, $f * g$, is also function in $\mathbb{C}^G$ and is defined by the values it takes at each $x \in G$ as follows:

$$(f * g)(x) = \sum_{y \in G} f(y) g(y^{-1}x). \tag{1}$$

Note that this is a weighted sum of translations of g. Indeed, let $\mathsf{T}_y : \mathbb{C}^G \to \mathbb{C}^G$ denote the *translation by y* operator—that is, T_y maps a function $g \in \mathbb{C}^G$ to a translated version of itself, $\mathsf{T}_y(g)$, which is defined at each $x \in G$ by $\mathsf{T}_y(g)(x) = g(y^{-1}x)$. Then (1) can be written as

$$(f * g)(x) = \sum_{y \in G} f(y)\mathsf{T}_y(g)(x), \tag{2}$$

a sum of weighted translations of g where the coefficients $f(y)$ are the weights, and $\mathsf{T}_y(g)$ is the function g "shifted" by y. (When G is the abelian group $\mathbb{Z}/n\mathbb{Z}$ with addition modulo n, we have $\mathsf{T}_y(g)(x) = g(y^{-1}x) = g(x-y)$, so in this case $\mathsf{T}_y(g)$ is literally g shifted by y units to the right.)

Equation (2) defines the convolution, $f * g$, by giving its value at each $x \in G$. Using the translation operator, however, we can define convolution "functionally," instead of element-wise, as follows:

$$f * g = \sum_{y \in G} f(y)\mathsf{T}_y(g) \tag{3}$$

(Pause to look at the right hand side of (3), and let it sink in that this is a function that takes arguments $x \in G$; compare with the right hand side of (2).)

This is fine, but it is also useful to think of (3) as f acting on g. Indeed, on the right hand side of (3) we have the operator $\sum_{y \in G} f(y)\mathsf{T}_y$ that maps the function g to the function $f * g$. But on the left hand side we have a binary operation $f * g$, written in infix notation, which doesn't jibe very well with this functional interpretation. So, instead of saying "the convolution of f and g", and writing $f * g$, we will say "the convolution **by** f **of** g," and write $\mathsf{C}(f)(g)$. In this way, we have the *convolution by f* operator:

$$\mathsf{C}(f) = \sum_{y \in G} f(y)\mathsf{T}_y, \tag{4}$$

which is a weighted sum of translation operators. The function $\mathsf{C}(f)$ takes other functions, like g, as its argument.

So, the functional types we have here are the following:

$$\mathsf{C} : \mathbb{C}^G \to (\mathbb{C}^G)^{\mathbb{C}^G}$$

Given $f \in \mathbb{C}^G$,

$$\mathsf{C}(f) : \mathbb{C}^G \to \mathbb{C}^G$$

Given $f \in \mathbb{C}^G$ and $g \in \mathbb{C}^G$,

$$\mathsf{C}(f)(g) : G \to \mathbb{C}$$

Or, in the notational style of a functional programming language like Scala:

$$\mathsf{C} : (G \Rightarrow \mathbb{C}) \Rightarrow ((G \Rightarrow \mathbb{C}) \Rightarrow (G \Rightarrow \mathbb{C}))$$

Given $f \in \mathbb{C}^G$,

$$\mathsf{C}(f) : (G \Rightarrow \mathbb{C}) \Rightarrow (G \Rightarrow \mathbb{C})$$

Given $f \in \mathbb{C}^G$ and $g \in \mathbb{C}^G$,

$$\mathsf{C}(f)(g) : (G \Rightarrow \mathbb{C})$$

2 Background: Finite Groups

TODO(wjd): *Decide whether this section should go in the appendix.*

This section summarizes the notations, definitions, and important facts needed below. The presentation style is terse since the goal of this section is to distill from the more general literature only those results that are most relevant to our application. The books [1] and [2] treat similar material in a more thorough and rigorous manner. Throughout, $\mathbb{C}$ denotes complex numbers, G an arbitrary finite group, and $\mathcal{L}(G)$ the collection of complex valued functions defined on G. Other notations for $\mathcal{L}(G)$ are $\mathbb{C}^G$ and $\{f : G \to \mathbb{C}\}$.

2.1 Cyclic Groups

A group C is called a *cyclic group* if there exists $x \in C$ such that every $y \in C$ has the form $y = x^n$ for some integer n. In this case, we call x a *generator* of C, and we say that such a group is *one generated*.

If G is an arbitrary finite group, and $x \in G$, then the set of powers of x,

$$\langle x \rangle = \{x^n : n \in \mathbb{Z}\}, \tag{5}$$

is a cyclic subgroup of G called the *group generated by x in G*.

It will be convenient to have notation for a cyclic group of order N without reference to a particular underlying group. Let the set of formal symbols

$$C_N(x) = \{x^n : 0 \leq n < N\} \tag{6}$$

denote the cyclic group of order N with generator x, and define binary composition by

$$x^m x^n = x^{m+n}, \quad 0 \leq m, n < N, \tag{7}$$

where $m + n$ is addition modulo N. Then $C_N(x)$ is a cyclic group of order N having generator x. The identity element of $C_N(x)$ is $x^0 = 1$, and the inverse of x^n in $C_N(x)$ is x^{N-n}.

To say that a group is *abelian* is to specify that the binary composition of the group is commutative, in which case the symbol $+$ is usually used to represent this operation. For nonabelian groups, we write the (non-commutative) binary composition as multiplication. Since our work involves both abelian and nonabelian groups, it is notationally cleaner to write the binary operations of an arbitrary group – abelian or otherwise – as multiplication. The following examples illustrate that additive groups, such as $\mathbb{Z}/N\mathbb{Z}$, have simple multiplicative representations.

Example. Let $\mathbb{Z}_N = \{0, 1, \dots, N-1\}$, and let addition modulo N be the binary composition defined on $\mathbb{Z}_N$. This group is isomorphic to the cyclic group $C_N(x)$,

$$\mathbb{Z}_N = \{n : 0 \le n < N\} \simeq \{x^n : 0 \le n < N\} = C_N(x),$$

and it is by this identification that the binary composition of $\mathbb{Z}_N$ can be written as multiplication. More precisely, by uniquely identifying each element $m \in \mathbb{Z}_N$ with the corresponding element $x^m \in C_N(x)$, the binary composition $m + n$ is replaced with that of (7).

Example. For an integer $\ell \in \mathbb{Z}_N$, denote by $\langle x^\ell \rangle$ the subgroup generated by x^ℓ in $C_N(x)$. If ℓ divides N, then

$$\langle x^\ell \rangle = \{x^{m\ell} : 0 \le m < M\}, \quad \ell M = N,$$

and $\langle x^\ell \rangle$ is a cyclic group of order M.

2.2 Group of Units

Multiplication modulo N is a ring product on the group of integers $\mathbb{Z}_N$. An element $m \in \mathbb{Z}_N$ is called a *unit* if there exists an $n \in \mathbb{Z}_N$ such that $mn = 1$. The set $U(N)$ of all units in $\mathbb{Z}_N$ is a group with respect to multiplication modulo N, and is called the *group of units*.The group of units can be described as the set of all integers $0 < m < N$ such that m and N are relatively prime.

For $N = 8$, $U(8) = \{1, 3, 5, 7\}$.

3 Translation Invariance

3.1 Generalized Translation and Convolution

For $y \in G$, the mapping T_y of $\mathcal{L}(G)$ defined by

$$(\mathsf{T}_y f)(x) = f(y^{-1}x), \quad x \in G, \tag{8}$$

is a linear operator of $\mathcal{L}(G)$ called *left translation by y*.

The mapping $\mathsf{C}(f)$ of $\mathcal{L}(G)$ defined by

$$\mathsf{C}(f) = \sum_{y \in G} f(y)\mathsf{T}_y, \quad f \in \mathcal{L}(G), \tag{9}$$

is a linear operator of $\mathcal{L}(G)$ called *left convolution by f*. By definition, for $x \in G$,

$$(\mathsf{C}(f)g)(x) = \sum_{y \in G} f(y)g(y^{-1}x), \quad g \in \mathcal{L}(G). \tag{10}$$

For $f, g \in \mathcal{L}(G)$, the composition $f * g = \mathsf{C}(f)g$ is called the *convolution product*. The vector space $\mathcal{L}(G)$ paired with the convolution product is an algebra, the *convolution algebra over G*.

To gain some familiarity with the general definitions of translation and convolution, it helps to verify that these definitions agree with what we expect when G is a familiar abelian group.

Example: If $G = \mathbb{Z}_N$, then (8) becomes

$$(\mathsf{T}_y f)(x) = f(x - y), \qquad x \in G, \tag{11}$$

and (10) becomes

$$(\mathsf{C}(g)f)(x) = \sum_{y \in G} g(y) f(x - y). \tag{12}$$

3.2 The Group Algebra $\mathbb{C}G$

The *group algebra* $\mathbb{C}G$ is the space of all formal sums

$$f = \sum_{x \in G} f(x) x, \quad f(x) \in \mathbb{C}, \tag{13}$$

with the following operations for $f, g \in \mathbb{C}G$:

$$f + g = \sum_{x \in G} (f(x) + g(x)) x, \tag{14}$$

$$\alpha f = \sum_{x \in G} (\alpha f(x)) x, \quad \alpha \in \mathbb{C}, \tag{15}$$

$$fg = \sum_{x \in G} \left(\sum_{y \in G} f(y) g(y^{-1} x) \right) x. \tag{16}$$

The mapping L_g of $\mathbb{C}G$ defined by $\mathsf{L}_g f = gf$ is a linear operator on the space $\mathbb{C}G$ called *left multiplication by g*. Since $y \in G$ can be identified with the formal sum $e_y \in \mathbb{C}G$ consisting of a single nonzero term,

$$yf = \mathsf{L}_{e_y} f = \sum_{x \in G} f(y^{-1} x) x. \tag{17}$$

In relation to translation of $\mathcal{L}(G)$, (17) is the $\mathbb{C}G$ analog. Fig. 1 illustrates.

The mapping $\Theta : \mathcal{L}(G) \to \mathbb{C}G$ defined by

$$\Theta(f) = \sum_{x \in G} f(x) x, \quad f \in \mathcal{L}(G), \tag{18}$$

is an algebra isomorphism of the convolution algebra $\mathcal{L}(G)$ onto the group algebra $\mathbb{C}G$. Thus we can identify $\Theta(f)$ with f, using context to decide whether f refers to the function in $\mathcal{L}(G)$ or the formal sum in $\mathbb{C}G$.

An important aspect of the foregoing isomorphism is the correspondence between the translations of the spaces. Translation of $\mathcal{L}(G)$ by $y \in G$ corresponds to left multiplication of $\mathbb{C}G$ by $y \in G$. Convolution of $\mathcal{L}(G)$ by $f \in \mathcal{L}(G)$ corresponds to left multiplication of $\mathbb{C}G$ by $f \in \mathbb{C}G$. We state these relations symbolically as follows:

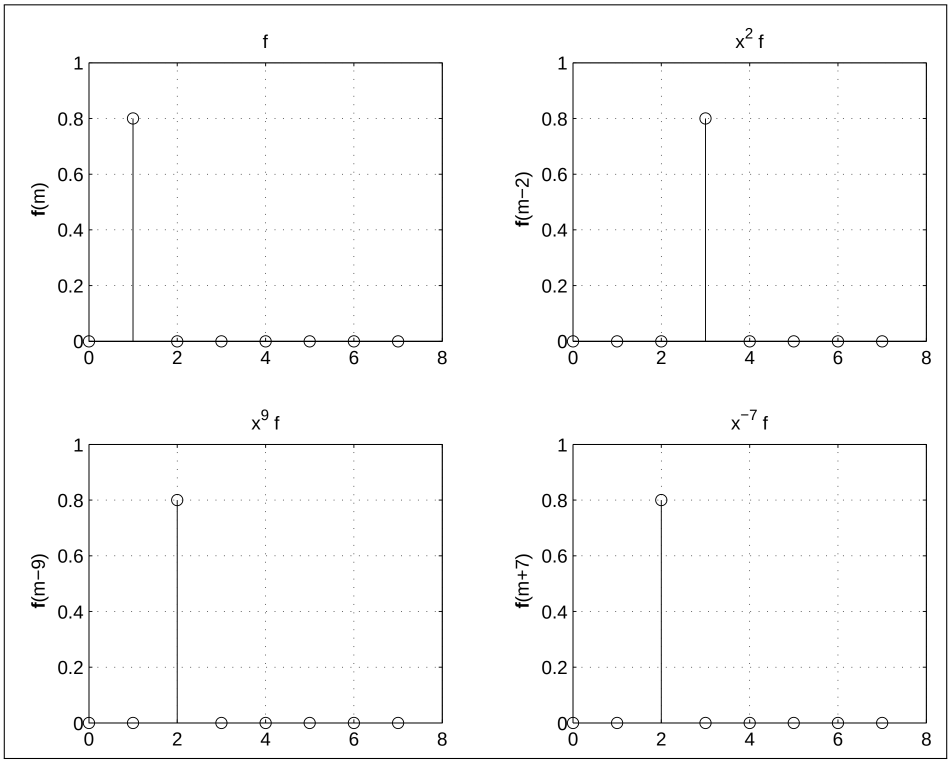

Fig. 1: An impulse $f \in \mathbb{C}A$ and a few abelian group translates, $x^2 f, x^9 f, x^{-7} f$.

$$\begin{array}{ccc} \mathcal{L}(G) & \simeq & \mathbb{C}G \\ \mathsf{T}_y & \leftrightarrow & \mathsf{L}_y \\ \mathsf{C}(f) & \leftrightarrow & \mathsf{L}_f \end{array}$$

4 Semidirect Product Groups

To determine whether a particular group is useful for a DSP application, we must specify exactly how the group indexes the data. An algorithm based on a given group may reduce computational complexity, it may make it easier to understand and perform a given signal processing task, or it may alter the results of a signal processing operation in an interesting or desirable way.

This section describes a simple class of nonabelian groups, called *abelian by abelian semidirect products*, that is useful in some applications. This class is perhaps the simplest extension of abelian groups, and consists of groups of the form $G = A \rtimes B$, where A and B are abelian groups. Not surprisingly, and DSP over such groups can be carried out just as it is done over abelian groups. However, the results of performing a given DSP operation can be vastly different, depending on which group is used as the underlying indexing set.

Let G be a finite group of order N, K a subgroup of G, and H a normal subgroup of G. If $G = HK$ and $H \cap K = \{1\}$, then we say that G is the *semidirect product* $G = H \rtimes K$. It can be shown that $G = H \rtimes K$ if and only if every $x \in G$ has a unique representation of the form $x = yz,\ y \in H, z \in K$.

Denote by $Aut(H)$ the set of all *automorphisms* of H. The mapping $\Psi : K \to Aut(H)$ defined by

$$\Psi_z(x) = zxz^{-1}, \quad z \in K, x \in H \tag{19}$$

is a group homomorphism. Define the binary composition in G in terms of Ψ as follows:

$$x_1x_2 = (y_1z_1)(y_2z_2) = y_1\Psi_{z_1}(y_2)z_1z_2, \tag{20}$$
$$y_1, y_2 \in H,\ z_1, z_2 \in K.$$

If K is a normal subgroup of G, then $y^{-1}Ky = K$ for all $y \in G$, and G is simply the cartesian product $H \times K$ with component-wise multiplication. What is new in the semidirect product is the possibility that K acts nontrivially on H. For this reason, K is sometimes called the "action group."

4.1 Simplest Nonabelian Example

If the mapping Ψ given in (19) is defined over $K = U(N)$, then Ψ is a group isomorphism. Under this identification, we can form the semidirect product $G = H \rtimes K$, with $H = C_N(x)$ and K a subgroup of $U(N)$. Throughout this section, G will denote such a semidirect product group.

The elements $u \in K$ are integers. However, following [1] we denote by k_u the element $u \in K$, as this avoids confusion that can arise on occasion. This notation is especially useful when K is a cyclic group with generator u. If we denote elements of K by k_u^j, instead of by u^j, it is easier to distinguish them from elements of the abelian group $C_N(x)$.

Suppose the action group K is a cyclic group of order $J = |K|$ with generator u. We identify each element of K with an index, and denote the set of elements by $K = \{k_u^j : 0 \leq j < J\}$. Thus, to each $k_v \in K$, there corresponds a $j \in \mathbb{Z}$ such that $k_u^j = k_v$. We use $x^n k_v$ and $x^n k_u^j$ to denote typical points of $G = C_N(x) \rtimes K$.

Given two points in G, say $z = x^m k_u$ and $y = x^n k_v$, define multiplication according to (20) as follows:

$$zy = (x^m k_u)(x^n k_v) = x^{m+un} k_u k_v, \tag{21}$$

where $m + un$ is taken modulo N. Since $k_v = k_u^j$ for some $j \in \mathbb{Z}$, then $k_u k_v = k_u^{1+j}$, and $zy = x^{m+un} k_u^{j+1}$.

Let $z = x^m k_v$ and suppose k_w is the inverse of k_v in K. Then the inverse of z must be $z^{-1} = x^{N-wm} k_w$, since this satisfies $z^{-1}z \equiv 1$.

Suppose $K \subset U(N)$ has order $|K| = J$, and consider the semidirect product group with elements

$$G = \{x^n k_u^j : 0 \leq n < N, 0 \leq j < J\}. \tag{22}$$

For $f \in \mathbb{C}G$,

$$f = \sum_{y \in G} f(y)y = \sum_{n,j} f(x^n k_u^j) x^n k_u^j, \tag{23}$$

As above, translations of $\mathbb{C}G$ are defined as left multiplication by elements of G. For semidirect product (22) there is a simple dichotomy of translation types that arise from left-multiplication by elements of G. First, the familiar “abelian translates” are obtained upon left-multiplication by powers of x (Fig. 1). By change of variables,

$$x^m f = \sum_{n,j} f(x^{n-m} k_u^j) x^n k_u^j, \tag{24}$$

which is simply a right shift of f by m units. Similarly, left-multiplication by powers of x^{-1} effects left shift of f. (Recall, $x^{-1} \equiv x^{N-1}$ and $x^{-m} \equiv x^{N-m}$.)

Of the second type are the “nonabelian translates,” obtained upon left-multiplication by $k_v \in K$.

$$k_v f = \sum_{n,j} f(k_v^{-1} x^n k_u^j) x^n k_u^j. \tag{25}$$

Suppose $k_w = k_u^\ell$ is the inverse of k_v in K. Then,

$$k_v f = \sum_{n,j} f(x^{wn} k_u^{\ell+j}) x^n k_u^j \tag{26}$$

From equation (26) it is clear that $k_v f$ results in a more complex transformation than that of $x^m f$ as given by (24).

For the general element $z = x^m k_v \in G$ with inverse $z^{-1} = x^{N-wm} k_w$ we derive rules for generalized translations.

$$zf = \sum_{y \in G} f(z^{-1}y)y = \sum_{n,j} f(x^{N-w(m-n)} k_w k_u^j) x^n k_u^j$$

$$z^{-1} f = \sum_{y \in G} f(zy)y = \sum_{n,j} f(x^{m+vn} k_v k_u^j) x^n k_u^j$$

To summarize, when different group are used as indexing sets, and products in the resulting group algebra are computed, interesting signal transforms result. In the next section, we elucidate the nature of these operations by examining some simple concrete examples in detail.

4.2 Examples: 1-D Semidirect Product Indexing Sets

Recall, the mapping $\Psi : U(N) \to Aut(C_N(x))$ is a group isomorphism. Under this identification, we can form $C_N(x) \rtimes K$ for any subgroup K of the group of units $U(N)$. A typical point in $C_N(x) \rtimes K$ is denoted (x^n, u), $0 \le n < N$, $u \in K$ with multiplication given by

$$(x^m, u)(x^n, v) = (x^{m+un}, uv), \quad 0 \le m, n < N,\ u, v \in K$$

where $m+un$ is taken modulo N. We often use k_u to denote the element $u \in K$ as this avoids confusion that can arise at various places.

Example 1. (See also [1, page 125].) Let G_1 be the abelian group

$$G_1 = C_{2N}(x) = \{x^n : 0 \leq n < 2N\}. \tag{27}$$

Let G_2 be the *dihedral group* with elements

$$\begin{aligned} G_2 &= C_N(x) \rtimes \{1, k_{N-1}\} \\ &= \{x^n k_{N-1}^j : 0 \leq n < N, 0 \leq j < 2\}. \end{aligned}$$

We order the elements of G_2 as follows:

$$\{1, x, \ldots, x^{N-1}, k_{N-1}, xk_{N-1}, \ldots, x^{N-1}k_{N-1}\}$$

Thus, G_2 is divided into two blocks with N-samples per block.

Example 2. Another group, G_3, will be constructed as follows: for some integer $M \geq 2$, define $N = 2^M$, so that $\left(\frac{N}{2} + 1\right)^2 \equiv 1 \mod N$, and $N/2 + 1$ generates a subgroup of $U(N)$ of order 2. Let

$$\begin{aligned} G_3 &= C_N(x) \rtimes \{1, k_{\frac{N}{2}+1}\} \\ &= \{x^n k_{\frac{N}{2}+1}^j : 0 \leq n < N, 0 \leq j < 2\}. \end{aligned}$$

Note that G_2 and G_3 are isomorphic groups.

Example 1 (cont.) By describing the translations of functions in $\mathbb{C}G_2$, we will see that the nonabelian translates of $\mathbb{C}G_2$ are "intrablock time-reversal" operations. A similar analysis of G_3 shows that the nonabelian translates of $\mathbb{C}G_3$ perform an "intrablock interleave" operation.

Multiplication on G_2 obeys the following relations:

$$x^N = k_{N-1}^2 = 1, \tag{28}$$

$$x^m k_{N-1}^{j+1}\, x^n k_{N-1}^j = \begin{cases} x^{m-n}, & j = 0, \\ x^{m+n}, & j = 1. \end{cases} \tag{29}$$

If $z = x^m k_{N-1}$, then $z^2 = 1$, thus $z^{-1} = z$.

For $f \in \mathbb{C}G_2$,

$$f = \sum_n f(x^n)x^n + f(x^n k_{N-1})x^n k_{N-1}. \tag{30}$$

By (28), the nonabelian translate $k_{N-1}f$ is given by

$$\sum_n f(k_{N-1}x^n)x^n + f(k_{N-1}x^n k_{N-1})x^n k_{N-1}$$

which is equivalent to

$$\sum_n f(x^{N-n}k_{N-1})x^n + f(x^{N-n})x^n k_{N-1}. \tag{31}$$

Comparing (30) and (31), we see that the nonabelian translate of $f \in \mathbb{C}G_2$ swaps the first N samples of f with the remaining N samples, and performs a time-reversal within each sub-block.

To express this another way, define $h = k_{N-1}f$. The first N coefficients of h are defined in terms of f as

$$h(x^n) = f(x^{N-n}k_{N-1}), \quad 0 \le n < N,$$

while the remaining N coefficients are given by

$$h(x^n k_{N-1}) = f(x^{N-n}), \quad 0 \le n < N.$$

For a simple linear function, this special translation is illustrated in Fig. 3.

A similar analysis of $G_3 = C_N(x) \rtimes \{1, k_{\frac{N}{2}+1}\}$ reveals that the nonabelian translates of $\mathbb{C}G_3$ interleave the elements within each N-sample sub-block of G_3, in addition to swapping the two blocks. This is illustrated in Fig. 4.

4.3 A Few Generalized Convolutions Computed

Fig. 2 illustrates a cyclic convolution of two discrete signals, with 16 samples each, indexed with the abelian group C_{16}. The first graph in Fig. 2 is a graph of the signal f, which is simply an impulse at the 9th sample; that is, $f(x^8) = 1$ and $f(x^m) = 0$, $m \neq 8$. The second signal, g, appears in the middle graph of Fig. 2. A linearly increasing sequence of 16 numbers ranging from -1 to 1, g can be represented as a vector of values

$$\mathbf{g} = (-1,\ -0.8\bar{6},\ -0.7\bar{3}, \ldots, 0.7\bar{3},\ 0.8\bar{6},\ 1) \tag{32}$$

or as an element of the group algebra $\mathbb{C}C_{16}$,

$$g = \sum_{m=0}^{15} g(x^m)x^m,$$

where the coefficients $g(x^m)$ take the values given in (32). The third graph in Fig. 2 shows the result of the convolution $\mathsf{C}(f)g = fg$. Evidently, when signals are indexed by elements of the abelian group C_{16}, then the product fg is the familiar cyclic convolution of f and g. (Recall, convolution by an impulse effects a translation.)

Fig. 3 shows the convolution $\mathsf{C}(f)g = fg$, where f is an impulse at the 9th sample, with group index k_7, and g is a linearly increasing sequence of 16 numbers ranging from -1 to 1; g can be represented as a vector, or as an element of the group algebra $\mathbb{C}(C_8 \rtimes \{1, k_7\})$,

$$g = \sum_{m=0}^{7} \sum_{j=0}^{1} g(x^m k_7^j)x^m k_7^j$$

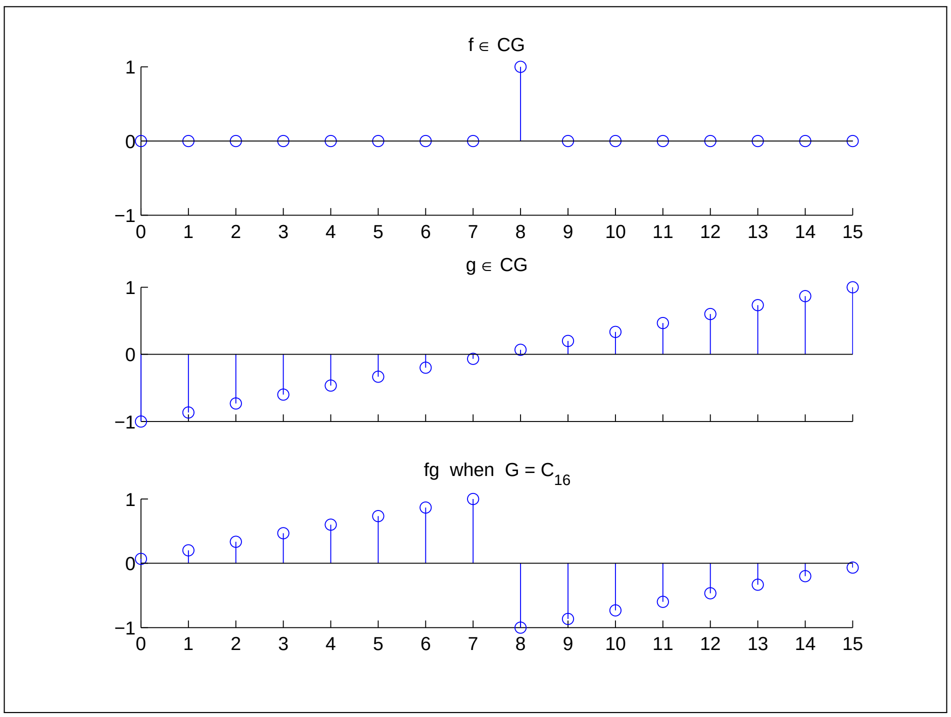

Fig. 2: Convolution of two signals indexed by the abelian group C_{16}.

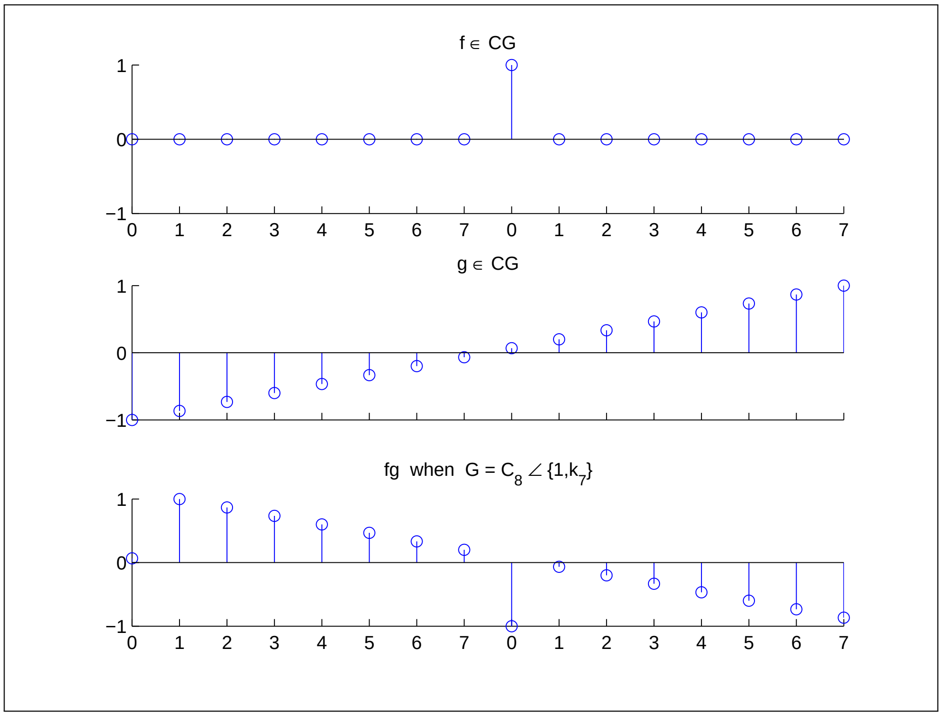

Fig. 3: The element $k_{N-1} \in G$ (top), where $N = 8$ and $G = C_8 \rtimes \{1, k_7\}$ – as an element of the group algebra, $f = k_7 \in \mathbb{C}G$ is the "impulse function" with one nonzero coefficient $f(k_7) = 1$; A linear signal $g \in \mathbb{C}G$ (middle); the product $fg = k_7 g$ (bottom) is, in general, the convolution of g by f, and is implemented by appealing to the convolution theorem and using a generalized FFT algorithm.

with coefficients $g(x^m k_7^j)$ taking the values given in (32); that is,

$$g(1) = -1,\ g(x) = -0.8\bar{6}, \ldots, g(x^7) = -0.0\bar{6},$$

$$g(k_7) = 0.0\bar{6},\ g(xk_7) = 0.2, \ldots, g(x^7 k_7) = 1.$$

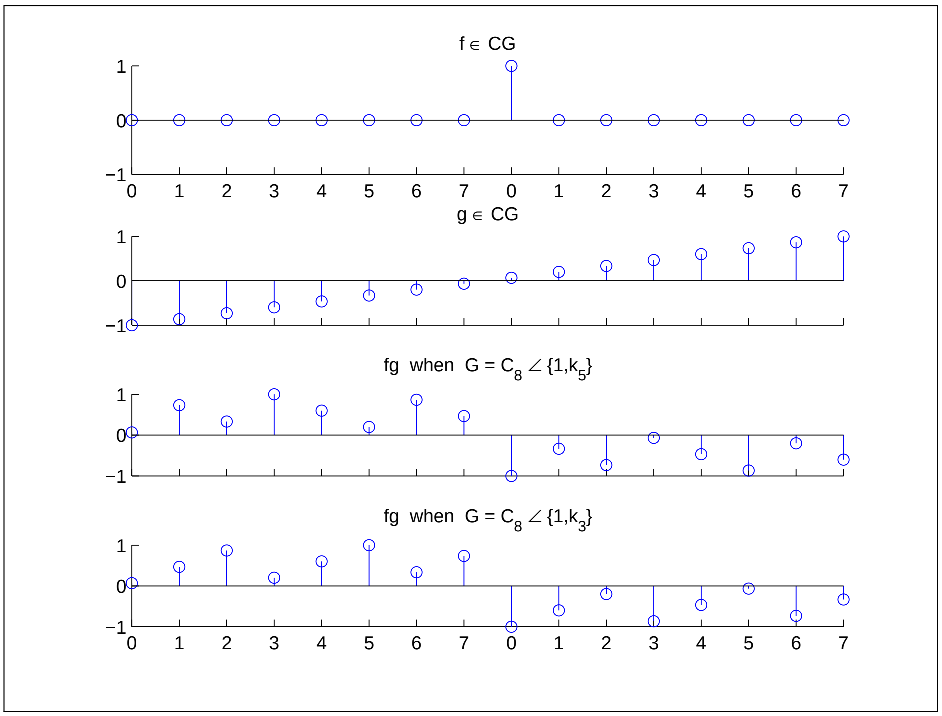

Fig. 4: Convolution of two signals indexed by the nonabelian groups $C_8 \rtimes \{1, k_5\}$ (third graph) and $C_8 \rtimes \{1, k_5\}$ (fourth graph).

Appendix

List of Acronyms

DSP digital signal processing

References

[1] Myoung An and Richard Tolimieri. *Group Filters and Image Processing*. Psypher Press, Boston, 2003. URL: http://prometheus-us.com/asi/algebra2003/papers/tolimieri.pdf.

[2] Richard Tolimieri and Myoung An. *Time-Frequency Representations*. Birkhäuser, Boston, 1998.

[3] Richard Tolimieri and Myoung An. *Group Filters and Image Processing*. Kluwer Acad., 2004.