# Audio Signal Representations for Factorization in the sparse domain

Manuel Moussallam, L. Daudet, G. Richard

# AUDIO SIGNAL REPRESENTATIONS FOR FACTORIZATION IN THE SPARSE DOMAIN

*Manuel Moussallam*[1,2] *, Laurent Daudet*[2]*, Gaël Richard*[1]

[1]Institut Telecom - Telecom ParisTech - CNRS/LTCI
37/39, rue Dareau 75014 Paris, France

[2]Institut Langevin - ESPCI ParisTech - UMR7587
10 rue Vauquelin, 75005 Paris, France

**ABSTRACT**

In this paper, a new class of audio representations is introduced, together with a corresponding fast decomposition algorithm. The main feature of these representations is that they are both sparse and approximately shift-invariant, which allows similarity search in a sparse domain. The common sparse support of detected similar patterns is then used to factorize their representations. The potential of this method for simultaneous structural analysis and compressing tasks is illustrated by preliminary experiments on simple musical data.

***Index Terms***— Sparse Representation, Matching Pursuit, Audio Signal Decomposition, Audio Similarity, Factorization

## 1. INTRODUCTION

The need for intelligent storage of ever-growing digital multimedia data has posed an interesting challenge on the Computer Science community. The rise of search engines, file-sharing plateforms and social networks on the internet, has proven that the ability to quickly retrieve consistent information from huge databases is now as important as managing its storage. One can see it as a joint optimization problem of space consumption and data access time but also as a concurrency between machine-oriented and human-readable systems. The compromise is set by the choice of a representation (i.e a transformation) of the data, that will hopefully be much sparser than the original data while keeping as much information as possible (ideally all of it), that is to say capturing only its very essence, and making it easily parsable by machines and/or humans.

A typical such task is archival of audio data, where indexing and compressing at the same time is crucial. Intuition (along with considerable research in the past years) tells us that a good way to proceed is to decompose the time-series that constitute the raw data in audio, as a combination of elementary sound *objects* or *atoms*. This idea is also strongly linked to the belief that human brain somehow processes sounds in a similar fashion, at least at a perception level [1].

So far, audio coding schemes have essentially focused on reducing short-term redundancies (usually frame-by-frame). The possibility of using long term redundancies, as can be done on text using sequential data compression methods [2] where repeating patterns are detected and factorized, remains a challenging issue for audio signals. However, pattern repetition is an extremely common feature in music, with groups of notes called "motifs" in classical music, or "riffs" in popular music. At a larger time-scale, the verse-chorus structure is another example of (near) repetition. In a radio station, the same popular song may be played tens of times during a single day. For archival purposes, one may be interested in audio representations where these repetitions, or near-repetitions, appear clearly.

Looking for large-scale similarities directly in the PCM domain is doomed to failure, first because the data rates would make this an extremely high-dimensional search problem, but more fundamentally because most of the time the repeats are not perfect at the bit level : note levels, durations, etc. may be slightly different. A better idea is to look for similarities in a sparse domain, for instance one of the transform-domain used in audio coding, e.g. the Modified Discrete Cosine Transform (MDCT). Recently, Ravelli [3] has enlightened the potential of a multiscale MDCT-based greedy decompositions for very low bit-rate coding : this larger dictionary allows for even sparser decompositions. Working in a sparse domain has substantial advantages: first, the amount of data is reduced, with hope that pattern classification becomes tractable again ; then, the information is sorted, with the largest coefficients accounting for the most salient sound structures : similarity search can then be progressive, first at a crude level, and then with finer and finer details.

However, these MDCT-based decompositions suffer from a major drawback : because they rely on some a priori slicing of the signal, they are not shift-invariant. In other words, the representation of a time-shifted signal $x(\cdot - \tau)$ can be quite different from the representation of $x$ with the time parameters shifted by $\tau$. All the timing (i.e., phase) information is here hidden in the amplitude of the MDCT parameters.

The main contributions of this paper can be summarized as follows :

- we propose a new shift-invariant decomposition (in the sense of [4]), based on adding time-shifts to the multiscale MDCT of [3],
- we provide an efficient decomposition scheme, called LOMP (for Locally Optimized Matching Pursuit), that effectively finds near shift-invariant decompositions at the cost of a small increase in computational complexity per iteration, compared to standard shift-dependent Matching Pursuit (note that this drawback is offset by a significantly faster energy decay, hence the addition of shift-invariance results in a *faster* algorithm per dB of SNR).
- we assess the potential of these decompositions for factorizing repeated musical patterns, on preliminary tests.

The rest of the paper goes as follows: Section 2 briefly introduces the multiscale MDCT Matching Pursuit framework and its observed limitations for our purposes. The novel shift-invariant decomposition, together with an effective decomposition algorithm, that reduces these limitations is presented in Section 3. Then, in Section 4, this new representation is applied to audio factorization and preliminary compression experiments illustrate its potential.

This work was partly supported by the QUAERO Programme, funded by OSEO, French State agency for innovation and by the DReaM project (ANR- 09-CORD-006) of the French National Research Agency CONTINT program.

## 2. SIGNAL MODEL

### 2.1. Sparse Representation problem

Let $x \in \mathbb{R}^N$ be a digital signal of length $N$. Let $\mathbf{\Phi} = \{\phi_k\}_{k=1..K}$ be a dictionary of $K$ waveforms. We are looking for the smallest combination of $I$ atoms from $\mathbf{\Phi}$ that minimizes a quadratic error $\epsilon = \|x - \sum_{i=1}^{I} \alpha_i \phi_i\|^2$. For very large signals such as for audio, the most widely-used approach is to use a greedy algorithm, that iteratively builds a representation by selecting at each step the best atom in the dictionary and subtracting it from the signal. Convergence is assured by the fact that the overall energy of the residual is strictly decreasing, but there is no guarantee of finding the global minimum. The basic version is called Matching Pursuit [5], and it has been extensively studied, modified and improved [6]. In practice, one has to stop the algorithm after a finite number of steps, usually when the approximation $\tilde{x} = \sum_{\imath \in I} \alpha_i \phi_i$ fulfills a fidelity depth criteria, expressed as a Signal-To-Residual Ratio (SRR) :

$$SRR = 10 \log \left( \frac{\|\tilde{x}\|^2}{\|x - \tilde{x}\|^2} \right) \quad (1)$$

The approximation is fully characterized by the vector $\alpha$ which has $I$ non zero coefficients. If the dictionary is redundant enough (i.e $K > N$), one might expect sparsity (i.e $I << N$) to arise at an acceptable fidelity (SRR) level.

### 2.2. Multiscale MDCT dictionary

The Multiscale MDCT dictionary as described in [3] consists in a union of 8 MDCT basis of different scales $\mathbf{\Phi} = \bigcup \mathbf{\Phi}_m$, distributed in a dyadic way. The multiscale nature of the dictionary provides a structural decomposition of the sounds, with harmonics (using large scales), transient parts (using short, well localized atoms) and speech or noise (using middle-sized atoms). Efficient implementations of the Matching Pursuit algorithm take advantage of the hierarchical structure of the dictionary, and the use of Fast Fourier Transform to achieve nearly real-time performances.

### 2.3. Similarity detection and time invariance

The issue of similarity detection in the sparse domain has been studied for example in [7]. However, our main goal is quite different, since we seek to use the similarity information as a tool for factorizing similar audio segments. Thus, an additional constraint on the representation is that it should have some robustness to time shifts (in Blumensath's sense [4] ) that may result from non-aligned frame slicings. Representations obtained with the multi-scale dictionary above are not time-invariant, as with any MDCT-based decomposition. Actually, the phase information is embedded in the $\alpha$ coefficients, and thus they greatly vary with temporal offsets. This is a serious obstacle to similarity detection, since two similar audio segments may have different sparse decompositions if they are not sample-aligned.

## 3. LOCAL OPTIMIZATION OF THE MATCHING PURSUIT

### 3.1. Shift-invariant multiscale MDCT dictionary

An effective yet computationally intensive solution to the above issue is to use a fully shift-invariant dictionary as in [4]. Let $\mathbf{\Phi}^G$ be a multiscale MDCT dictionary, and let $\mathbf{\Phi}_m^G$ be a single scale sub-dictionary of size $L_m$. The MDCT is a 50% overlapped transform, therefore, for a $N$-length signal, $\mathbf{\Phi}_m^G$ is an orthonormal basis of $\mathbb{R}^{2N}$

$\mathbf{\Phi}_m^G$ can be upgraded in the following manner: for each atom $\phi_{m,i}$ of $\mathbf{\Phi}_m^G$, add all possible shifted versions $\phi_{m,i,\tau} = \phi_{m,i} * \delta_\tau$ with $\tau$ living in $\left[-\frac{L_m}{4}, \frac{L_m}{4}\right]$ to the dictionary, thus yielding $\mathbf{\Phi}_m^F$ a highly redundant dictionary of size $N.L_m$. Repeating this operation in all the basis, a global dictionary $\mathbf{\Phi}^F = \bigcup \mathbf{\Phi}_m^F$ can be built.

### 3.2. Hybrid approach: Locally Optimized Matching Pursuit (LOMP)

Using $\mathbf{\Phi}^F$ leads to sparser, clearer representations, but the complexity of a Matching Pursuit over $\mathbf{\Phi}^F$ (called here FullMP) would be prohibitive for real data analysis. We propose here a hybrid approach that uses atoms from $\mathbf{\Phi}^F$ at the cost of a search over $\mathbf{\Phi}^G$. Alternatively, this can be seen as an attempt to extract the phase information out of $\alpha$ into a separate vector $\tau$, thus yielding phase-invariance properties. At iteration $n$ let $r^n$ be the residual, the best atom $\phi_{k_{max}}^G = \arg\max_{\phi_k \in \mathbf{\Phi}^G} |\langle \phi_k, r^n \rangle|$ of length $L_m$ is selected in $\mathbf{\Phi}^G$. Then a local optimization is performed yielding

$$\phi_{k_{max},\tau_{max}}^F = \arg \max_{\tau \in [-\frac{L_m}{4}, \frac{L_m}{4}]} | \left\langle r^n, (\phi_{k_{max}}^G * \delta_\tau) \right\rangle | \quad (2)$$

where $*$ denotes the convolution product and $\delta_\tau$ is a discrete dirac impulse located in $\tau$ (this convolution is simply a time shift by $\tau$ samples). The modified algorithm is given in Algorithm 1, the major difference with the standard MP algorithm is summarized in step 4 in bold. It is worth noticing that contrary to the method proposed in [6], where smaller sub-atoms are constructed at each iteration to fit locally the energy distribution, the LOMP algorithm performs a temporal realignement of the selected atom, which can be implemented efficiently using FFTs.

**Algorithm 1** Locally Optimized Matching Pursuit (LOMP)

**Input**: $x$ , $\mathbf{\Phi}$
**Output**: $\alpha$, $\tau$ and $r$ such that $x = \sum_{i \in I} \alpha_i.(\phi_i * \delta(\tau_i)) + r$

1: $r0 := x$
2: **repeat**
3: Find index $k_{max} = \arg\max_{\phi_k \in \mathbf{\Phi}} | \langle \phi_k, r^n \rangle |$
**4:** **Compute local optimal time-shift**
$\tau_{max} = \arg\max_\tau | \langle r^n, (\phi_{k_{max}} * \delta_\tau) \rangle |$
5: Update projection score : $\alpha_{max} \leftarrow \langle r^n, \phi_{k_{max},\tau_{max}} \rangle$
6: Update residual : $r^{n+1} \leftarrow r^n - \alpha_{max}.\phi_{k_{max},\tau_{max}}$
7: **until** a stopping condition is met

#### *3.2.1. Convergence discussion*

As a weak (or partially) greedy algorithm, one can prove (see for instance [8]) the convergence of Algorithm 1. Moreover, one might easily be convinced that, at least for the first iterations, the convergence rate of the FullMP algorithm is faster than standard MP on the original (and much smaller) MDCT dictionary (called here "MP"). For LOMP algorithm, an intermediate behavior can be expected.

Figure 1 offers an insight of this for a short segment of orchestra music taken from the MPEG SQAM database. Implementation of all three algorithms using Python programming language showed an equivalent complexity for MP and LOMP, on a standard PC. Indeed, at each iteration : normalized by the SRR gain, LOMP has a smaller computation time than the standard (non shift-invariant) MP.

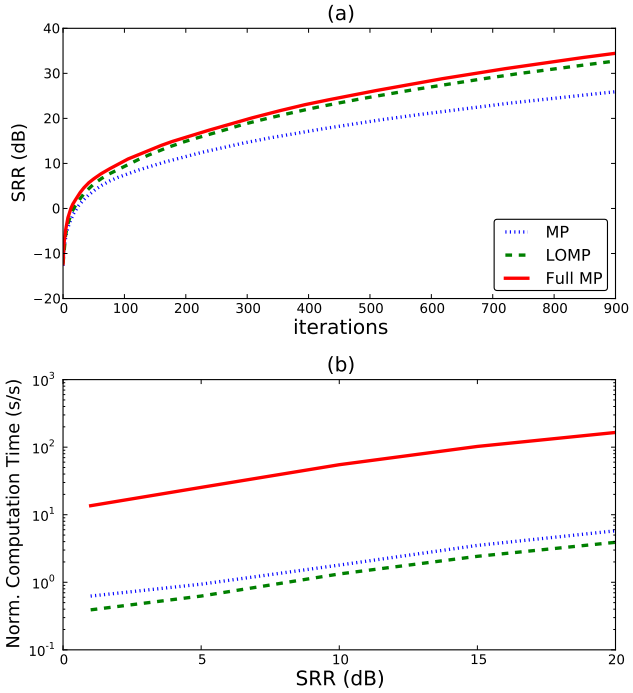


**Fig. 1**. *Comparison of (a) : convergence rates and (b) : Complexities in computational time , for MP , FullMP and LOMP decomposition of 4096 samples of orchestra signal over a dictionary of 3 MDCT basis (32,128,512).*

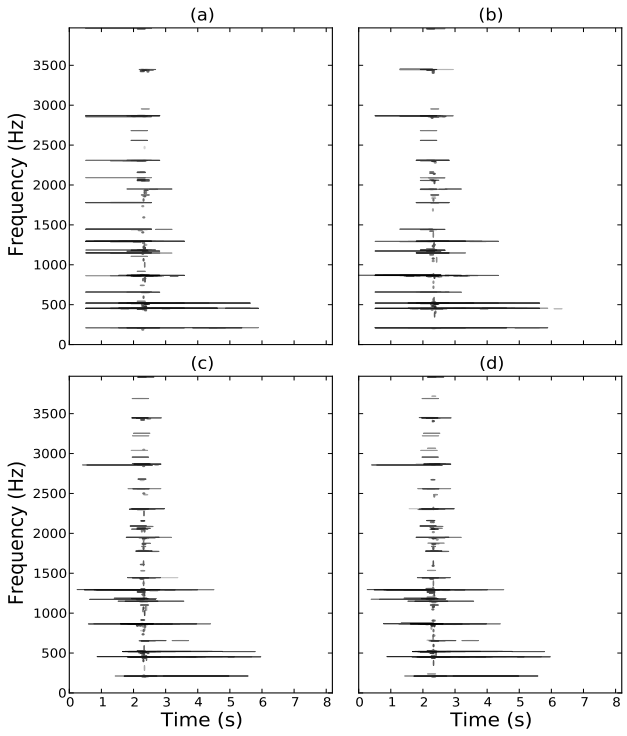


**Fig. 2**. *Energy distributions of a bell sound decomposed by MP and LOMP algorithms. Left: original signal decomposition at 20 dB of SRR with 8 MDCT basis with MP (a) and LOMP (c). Right: decomposition of the same signals shifted in time by 100 samples with MP (b) and LOMP (d).*

#### 3.2.2. *Time invariance discussion*

LOMP atoms are more expensive to encode since an additional time-shift coefficient is to be encoded. However, since the convergence rate is often faster, one might hope to obtain sparser representation. Section 4 will further investigate this point. Meanwhile, LOMP representations are much more robust to time shifts than MP ones. Figure 2 illustrates the lack of stability of MP decompositions and how LOMP solves this issue. Moreover, one can see that LOMP decomposition are sparser, and that temporal reassignment result in fewer pre-echo artefacts. Indeed, changes are visible between (a) and (b) due to phase shifts and different frame slicing, while (c) and (d) are almost perfectly identical. Such robustness paves the way for similarity detection and compressing redundant patterns in the sparse domain.

## 4. FACTORIZATION IN THE SPARSE DOMAIN

### 4.1. Factorization paradigm

Let $\tilde{x}$ and $\tilde{y}$ be the two LOMP representation in Figure 2 (c) and (d). The set of atoms $\{\phi_i\}$ and coefficient vector $\alpha$ for $\tilde{x}$ and $\tilde{y}$ are almost equals. In other words, both representation share the same sparse support and their differences are entirely characterized by the sets of time-shifts $\tau^x$ and $\tau^y$. This has two important consequences: first the common sparse support denotes a similarity between $x$ and $y$. Second, it means that one only needs to encode $\alpha$ and the set of atoms $\{\phi_i\}$ once, as a common factor to describe both $\tilde{x}$ and $\tilde{y}$. Alternatively, $\tilde{y}$ can be described using the sparse support of $\tilde{x}$ and $\tau^y$. Now this paradigm can be extended to any $x$ and $y$. Knowing a representation $\tilde{x}$ as a reference, let $\hat{y}$ be an approximation of $y$, which support is the same as $\tilde{x}$ and has a set of time-shift $\tau^y$. A method to build such $\hat{y}$ is described by Algorithm 2

**Algorithm 2** Factorization

**Input**: $y$ , $\mathbf{\Phi}$ , $\tilde{x} = \sum_{i \in I} \alpha_i \phi_i * \delta(\tau_i^x)$
**Output**: $\tau^y$ such that $y = \sum_{i \in I} \alpha_i \phi_i * \delta(\tau_i^y) + r$

1: $r^0 := y$
2: **for all** $i \in I$ **do**
3: Compute local optimal time-shift
$\tau_{max} = \arg\max_\tau |\langle r^n, (\alpha_i.\phi_i * \delta_\tau)\rangle|$
4: Update residual : $r^{n+1} \leftarrow r^n - \alpha_i.(\phi_i * \tau_{max})$
5: **end for**

This paradigm is close to the information theory concept of distributed source coding where repetitions are considered as correlated sources that are not co-located [9]. However, in audio and musical streams, a prior detection of these repeating segments is required.

### 4.2. Similarity Detection

If $x$ and $y$ are either identical or simply similar (which might be the case for repeated choruses in a pop song, chords sequences in classical music, or even whole songs and news reports on radio broadcast recordings), then chances are that atoms from $\tilde{x}$ are useful to de-

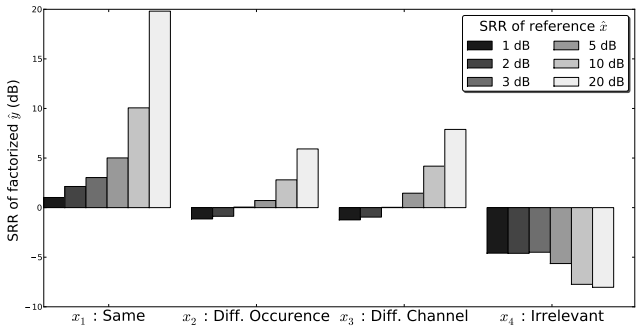


**Fig. 3**. *SRR reached by factorizing $y$ (few piano notes) using 4 different signals $x_i$ as references, at 5 different levels of decomposition*

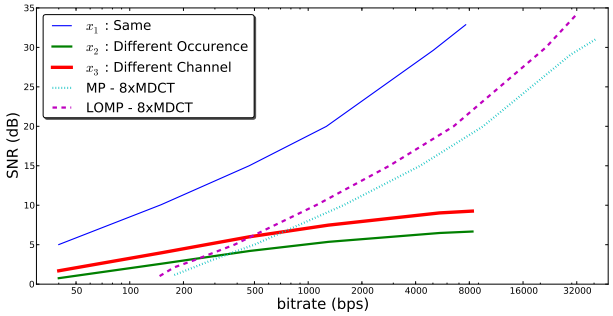


**Fig. 4**. *Bitrates and SNR obtained with MP, LOMP or factorization using different references*

scribe $y$, and a good way to measure it is to observe the residual's energy decay in Algorithm 2, or alternatively the SRR of reconstructed $\hat{y}$.

To illustrate this method, Figure 3 presents the SRR of $\hat{y}$ when using 4 different $\{x_i\}_{i=1..4}$ as reference signal for the factorization, and at different decomposition levels. Here $y$ is the first 8 piano notes (2 s, 1st half of the 1st bar) taken from the left channel of a piano recording of Bach's first Prelude in C Major, from the Well-Tempered Clavier. The first case is simply $x_1 = y$, obviously the factorization is complete and the reached SRR matches the reference's one. More interestingly, $x_2$ is the exact same musical pattern played right afterwards in the piece (2nd half of the 1st bar), thus the correlation between $x_2$ and $y$ is far lower. Third example, $x_3$ is the right channel corresponding to $y$. Finally, for the irrelevant case, $x_4$ is a few seconds exctracted from an orchestra recording, therefore having no similarity with $y$ .

One can observe that a factorization at a shallow depth (3 to 5 dB) allows us to detect the three types of similarity and reject the irrelevant case. Moreover, the detection process yields a preliminary decomposition of $y$ in the same time that is reveals the signal structure.

### 4.3. Compression Experiments

Knowing its reference $\tilde{x}$, a factorized approximation $\hat{y}$ of $y$ is described completely by the set of atom time shifts $\tau^y$. One can picture the factorization process as a greedy decomposition in which the sequence of atom choice in the dictionary and their amplitude is arbitrarily fixed (only local time-shifts are allowed). This strong constraint can result in a lower approximation precision compared to MP decomposition, but its cost is also fairly reduced, which means that high compression levels can be achieved.

Working on the same Bach's signal, different decomposition techniques are compared: MP and LOMP on a 8xMDCT dictionary, and factorization using LOMP decompositions of $x_1$, $x_2$ and $x_3$, in a simple encoding task. For MP and LOMP decompositions, atom coefficients are encoded using a simple 7-bit uniform scalar quantizer and plain entropy coding of the quantized coefficients, atom indexes have a fixed cost of $\log_2(2.N)$ and for LOMP atoms, an additional cost for the time shifts is also evaluated by plain entropy coding. For factorized representations, only the set of time shifts is encoded.

Results are shown Figure 4. The quality of the compression is here measured by the Signal-To-Noise Ratio (SNR) of the decoded signal. At very low bitrates, the factorization approach is interesting, even when the reference is a different (albeit closely related) signal. When near exact redundancy is detected and used for factorization, important coding gains may be achieved with this technique.

## 5. CONCLUSION AND FUTURE WORK

We have presented a multiscale greedy decomposition of audio signals that introduces robustness to time shifts, while increasing the sparsity of the representation, at a negligible computational penalty. Using this framework, similarity detection can be performed along with an enhanced compressive scheme through factorization in the representation domain, even for non-perfect redundancies. Many extensions of this work are yet to be undertaken, among which a study of the semantic similarities that may or may not be factorized. The scalability of this approach will also be investigated so as to design a complete audio coding scheme based on this model.

## 6. REFERENCES

[1] M. Lewicki, "Efficient coding of natural sounds," *Nature Neuroscience*, vol. 5(4), pp. 356 – 363, 2002.

[2] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inf. Th.*, vol. 23(3), pp. 337–343, 1977.

[3] E. Ravelli, G. Richard, and L. Daudet, "Union of MDCT bases for audio coding," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 16(8), pp. 1361–1372, 2008.

[4] T. Blumensath and M. Davies, "Sparse and shift-invariant representations of music," *IEEE Trans. Audio Speech Lang. Proc.*, vol. 14(1), pp. 50–57, 2006.

[5] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Sig. Proc.*, vol. 41(12), pp. 3397–3415, 1993.

[6] R. Gribonval, E. Bacry, S. Mallat, P. Depalle, and X. Rodet, "Analysis of sound signals with high resolution matching pursuit," *IEEE Symp. TFTS*, pp. 125–128, 1996.

[7] L. B. Sturm, "On similarity search in audio signals using adaptive sparse approximations," *Workshop on Adaptive Multimedia Retrieval, Madrid*, 2009.

[8] V.N. Temlyakov, "A criterion for convergence of weak greedy algorithms," *Adv. Comp. Math.*, vol. 17(3), pp. 269–280, 2002.

[9] S.S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): design and construction," *IEEE Trans. Inf. Th.*, vol. 49(3), pp. 626–643, 2003.