More on Morphological Mutation Functions
Recent Techniques and Developments

Larry Polansky
Bregman Electro-Acoustic Music Studio, Dartmouth College, Hanover, NH 03755
email: larry.polansky@mac.dartmouth.edu

Abstract: This paper describes recent work in the theory of morphological mutation functions: functions which transform, in a wide variety of ways, one morphology (shape) into another. This paper extends and refines ideas presented in Polansky and McKinney [1991]. Topics include: a simpler notation for the mutations, the concept of *zero-override*, a brief discussion of the notions of *generalized interval*, *clumping*, and *harmonic mutations*.

Morphological mutation functions are of the general form

$$f(S,T,\Omega) = M$$

where S and T are two morphologies, M is the resultant *mutant*, and Ω is the mutation index ranging from 0 and 1. These functions are described in detail in Polansky and McKinney [1991] in terms of their use in the transformation of melodic information, and in time- and frequency-domain cross synthesis. In general, the mutation functions deal with two types of intervallic information: *contour* and *magnitude*. All of the mutations described in [Polansky and McKinney] separate and then recombine these two morphological characteristics from S and T to create a third morphology, M, which is some kind of "cross-fade." The *degree* of cross-fade is determined by Ω.

This brief paper is a companion piece to Polansky and McKinney, and presents recent ideas, refinements of previous concepts, and suggestions for further experimentation. Many of these ideas were developed in the application of the mutation functions to two recent computer-composed compositions: *51 Melodies* and *Two Children's Songs* [Polansky, 1991, 1992].

Simplified notation

The mutations described in Polansky and McKinney may be notationally simplified to reflect various combinations of contour and magnitude and facilitate software implementation. Contour is described as the 3-valued sign of an interval {-1, 0, 1}. In this case, magnitude is the absolute value of the difference of two intervals.[1] The sign and magnitude of an interval between any two elements (i, j) in a morphology (S) is given by:

$$|S_i - S_j| = S_{mag} \text{ — and — } \frac{S_i - S_j}{|S_i - S_j|} = S_{sgn}$$

The mutation functions from [Polansky and McKinney] may be more simply rewritten as the following, where S and T are the two morphologies and M is the mutant:

- Linear Contour Mutation (LCM)
  - $M_i = M_j + T_{sgn} * S_{mag}$ (for mutated intervals)
  - $M_i = M_j + S_{sgn} * S_{mag}$ (general form for non-mutated intervals)[2]
- Irregular Unsigned Interval Magnitude (IUIM
  - $M_i = M_j + S_{sgn} * T_{mag}$ (for mutated intervals; non-mutated intervals as above)[3]
- Irregular Signed Interval Magnitude (ISIM)
  - $M_i = M_j + T_{sgn} * T_{mag}$ (for mutated intervals; non-mutated intervals as above)[4]
- Uniform Unsigned Interval Magnitude (UUIM)
  - $M_i = M_j + S_{sgn} * (S_{mag} + \Omega * |T_{mag} - S_{mag}|)$
- Uniform Signed Interval Magnitude (USIM)
  - $M_i = M_j + (S_{int}) + \Omega * (T_{int} - S_{int})$
  - — where S_{int} and T_{int} are $(S_{sgn} * S_{mag})$ and $(T_{sgn} * T_{mag})$ respectively
- Value Mutation (VM)
  - $M_i = S_i + \Omega * (T_i - S_i)$[5]

Zero-Override

One of the most common ways to use these mutations is by *concatenation*. For example, the concatenated interval *IUIM(LCM)* with $\Omega = 1$ will completely mutate S into T. Many different transformation trajectories are possible because of the different (and complementary) ways that the *IUIM* and *LCM* separate sign and magnitude for individual intervals [Polansky and McKinney]. A theoretical difficulty arises when either sign or magnitude of S_{int} is zero: the distinction

between sign and magnitude is confused. Example 1 and 2 show two instances of the *IUIM(LCM)* concatenation mutation (using linear adjacency intervals). In the first example there is no zero interval in the source, in the second there is.

Example 1 (no problem):

1) S: {7,8,3,2,1} *T:* {7,5,9,10,12}
2) LCM(S, T), Ω =1:{7,6,11,12,13}
$[M_i = M_J + T_{sgn} * S_{mag}]$
3) IUIM(LCM(S, T), T), Ω =1:{7,5,9,10,12} = *T*
$[M_i = M_J + S_{sgn} * T_{mag}]$, where *S=LCM(S, T)*

Example 2 (problem):

1) S: {7,7,3,2,1} *T:* {7,5,9,10,12}
2) LCM(S, T), Ω =1:{7,7,11,12,13}
(Down(*T*), by 0(*S*): (sign of *T* is lost because of zero magnitude of *S*)
3) IUIM(LCM(S, T),T), Ω =1:{7,7,9,10,12}$\neq T$

The same problem occurs in the *LCM(UUIM)*, *LCM(IUIM)*, and *UUIM(LCM)* mutations when there is no change between two elements in *S*. It is not possible to completely "arrive" at *T* because of the confusion of sign and magnitude in the inner function. A simple and practical solution is to slightly modify interval calculations that separate sign and magnitude: test for a zero interval in *S*, and if zero, use either the magnitude or sign of *T* for that interval (depending on the type of mutation). This is called *zero-override*. For longer morphologies, especially sound samples, zero-override does not cause much of a deformation in the mutation. Current versions of the software, on both the NeXT machine synthesis platform and in HMSL, allow for enabling or disabling zero-override, since the user may be more concerned with the integrity of the equations than in completely arriving at *T*. The following example shows the use of zero-override for the *LCM* mutation:

LCM: $M_i = M_j + T_{sgn} * S_{mag}$; where $S_{mag} \neq 0$
LCM: $M_i = M_j + T_{sgn} * T_{mag}$; where $S_{mag} = 0$ and ZERO-OVERRIDE is ON (do not mutate)

This simple procedure can be used for the *LCM*, *IUIM* and *UUIM* in the current software, and in two recent works, *51 Melodies* and *Two Children's Songs* [Polansky, 1991; 1992].

Generalized interval calculations

In the actual interval calculations, a high degree of generality is desired for maximum possibility of experimentation. The mutation software (in HMSL) is written so that a wide variety of *interval types* are selectable by the programmer. Obviously, arithmetic, ratiometric and several types of contour intervals (for example, ternary and binary [Polansky and Bassein], but also perhaps quintary and higher contour "grains") need to be available for the definitions of the various mutation functions. In addition, the specification of the particular *indices* used in interval calculations is quite general.

Combinatorial interval calculations are in general either calculations between non-adjacent indices, or calculations which involve greater than *L-1* intervals (where *L* is the length of a morphology) [Polansky, 1987]. Combinatorial mutations (and metrics) are obviously more sensitive than linear ones, and will restrict the "wandering DC offset" phenomenon caused by mutations like the *LCM*. Polansky and McKinney used a high-pass filter to correct this problem in time-domain synthesis applications. Combinatorial magnitude and linear magnitude *metrics* are in general equivalent, unless rows of the combinatorial matrices are weighted differently, which is not the case for contour.[6] Different definitions of *adjacency* in the calculation of intervals produce different forms of combinatoriality. For example, if the *adjacency index* is a variable, then certain morphologies may calculate intervals on the basis of every second, third, or fourth interval. This may differ for *S* and *T*, and the adjacency may change over the course of a mutation.

The distinction between *fundamental*, *external fundamental* and *fundamental index* has proved extremely useful. A *fundamental* is a value to which all intervals are computed. In *51 Melodies*, the note E is used as the fundamental for all pitch mutations, which kept the piece more or less related to a certain tonality, and within the guitar's range. If the fundamental does not actually occur in the morphology, it is called *external*. For example, in mutating a melody, the mutation might be "around" the note C, where that note never appears in the melody. *Fundamental mutations* are an excellent way of normalizing the absolute value of a mutation. For example, in time- or frequency-domain soundfile mutations, taking all intervals to zero can be more effective than calculating adjacent intervals.

A *fundamental index* is the specification of an index, not a value, to which interval calculations take place: what is desired is the relationship to a particular *index* in the morphology, not any specific *value*. This is especially useful when the index itself changes over the course of a mutation. As in the case of adjacency intervals, different fundamentals, external or not, may be specified for *S* and *T* for interesting results.

Clumping

> "Although the index in the *contour mutation* function determines what percentage of intervals get mutated, it does not determine exactly which intervals get mutated. ... There are many ways the mutated intervals can be distributed throughout the resultant morphology. The *clumping value* ∂ of a mutation function determines what percentage of the mutated intervals are mutated *consecutively*. ∂ ranges from 0.0 to 1.0, where a value of 0.0 spreads the mutated intervals uniformly across the mutated morphology and a value of 1.0 groups all of the mutated intervals into one clump of mutated intervals. For example, with $\Omega = .5$ and $\partial = 1.0$, the first half of the morphology would be mutated (clumping is as high as possible), the second half left unchanged." [Polansky and McKinney].

The concept of *clumping* may be thought of as a simple probability, p_i, of a value in the morphology being mutated or not, dependent on the two variables ∂ and Ω.[7] It is important that the specification of ∂ be independent of Ω so that the number of changes can be specified irrespective of the distribution (in particular the "clumping") of the changes. Clearly where $\Omega = 1$, all values need to be mutated for all values of ∂ ($p_i = 1$ for all i), and for $\Omega = 0$, no values should be mutated ($p_i = 0$ for all i). For intermediate values, like $\Omega = .5$, ∂ specifies the "spread" of the mutations. So for example, with $\partial = 1$, the mutations should be maximally clumped.[8]

The clumping equation:

$$p_i = \Omega^{1-\partial}$$

satisfies these criteria.[9] As $\partial \to 0$, $p_i \to \Omega$: with "no" clumping, the probability is simply the mutation index. This is equivalent to stochastic clumping, described in Polansky and McKinney. As $\partial \to 1$, so does p_i (irregardless of Ω). This means that with a high clumping value, the probability function "tries" to "always change the next value" (with no statistical feedback mechanism). The following shows the "ranges" of the equation:

(1) $\partial = 0, \Omega = 0, p =$ *undefined*, set to 0 (approaches 0 as limit)
(2) $\partial = 1, \Omega = 0, p = 0$
(3) $\partial = 1, \Omega = 1, p = 1$
(4) $\partial = 0, \Omega = 1, p = 1$

Note that the equation does not guarantee that the resulting mutant will have the desired ($\Omega * L$) number of mutated indices. It is simply an index-by-index probability function which produces the clumping effect. Intuitively, however, Ω should reflect, more or less, the number of indices actually changed. As in the case of most probability distributions, this can be achieved by statistical feedback [Ames 1991a, 1990, 1986; or see Casey, 1992 for an elegant formulation]. The simplest form of statistical feedback, in this case, is to repeatedly test the number of mutations done against the number of mutations desired, and stop when that number is reached. This takes care of the case where Ω is low and ∂ is high. This primitive feedback mechanism ensures that the number of mutations does not exceed ($\Omega * L$), but does not guarantee that the "desired" distribution will occur, especially for "odd" relationships of Ω and ∂. For example, for $\Omega = .5$ and $\partial = 0$, the "desired" distribution can be thought of as "every other one," or $\Omega = .5$, $\partial = 1$, as the "first half of the morphology" and so on. In fact, using this equation, complex relationships between Ω and ∂ (e.g $\Omega = .5$, $\partial = .75$) become an interesting area for further experimentation. This equation, although developed to solve a particular problem in the mutation of morphologies, may also have more general musical applications, in either time- or frequency-domain cross synthesis, or at a higher level.

Harmonic mutations

In *Two Children's Songs*, a recent work for tuba and trombone (or any two bass winds), I used a new mutation function called a *harmonic mutation*. This mutation crossfades between harmonic distance values of two morphologies, and then "de-references" those values to find the correct pitch. The composer assigns harmonic distance values for a given morphology or piece. For example, if a perfect fifth is assigned the harmonic value of 1 (least distant, with the octave being 0), and the tritone a value of 11 (most distant), $\Omega = .5$ would yield a value of 5, which might be, for example, a major sixth. In other words, in the key of Bb, this would take an F in *S*, and an E-natural in *T*, and yield a G in *M*. The harmonic mutations also allow for the separation of octave and pitch class, in a manner analogous to the separation of magnitude and contour in the other mutations. In this way, either octave or "harmonic distance" may be mutated independently, and of course concatenated. In addition, all of the generalized interval concepts are retained in these mutations, so that harmonic values might be computed from note to note, to some sense of "tonic", or to some shifting, dynamically computed value, and of course can be defined differently for *S* and *T*. This type of "table-lookup" distance computation might be useful for other musical parameters as well (rhythm, timbre, etc.), and is an interesting area for further work. These harmonic mutations might be extended to incorporate more sensitive harmonic distance functions (at present, each interval is simply ranked from 0-11), and could be used for more sophisticated harmonic spaces (other than 12-tone equal temperament).

Future work and recent implementations

Recent work on the mutation functions has centered on the NeXT machine synthesis platform and the HMSL implementation for higher level morphologies. Eric Smith has extended Martin McKinney's cross-synthesis platform using the NeXT Interface Builder and incorporated several of the ideas mentioned above. With Phil Burk, I havegreatly expanded and refined the HMSL mutation software to include: harmonic mutations, several of the ideas described above, and a real-time graphic interface for experimentation with the various mutation parameters. This graphic interface allows the user to enter in melodies (S and T) via some MIDI device, select the type of mutation, the dimension of mutation (pitch, loudness, or duration), Ω, and other parameters, and then hear the mutant morphology in real-time. Tom Erbe, at the Mills College Center for Contemporary Music, is currently incorporating the frequency-domain mutations into his SoundHack program. All of the above programs are distributed freely.

References

Ames, Charles. 1991. "A Catalog of Statistical Distributions: Techniques for Transforming Random, Determinate and Chaotic Sequences." *Leonardo Music Journal*. Volume 1, Number 1.

— 1992, 1991a. Several personal communcations concerning the notion of clumping.

— 1990. "Statistics and Compositional Balance." *Perspectives of New Music*. Vol. 28. No. 1. Winter.

— 1986. "Two Pieces for Amplified Guitar." *Interface*. Vol. 15, No. 1.

Casey, Michael. 1992. "HS: A Symbolic Programming Language for Computer Assisted Composition." Master's Thesis, Dartmouth College.

McKinney, Martin. 1991. "Mutation Synthesis." Master's Thesis, Dartmouth College.

Polansky, L. 1991. *51 Melodies*. Score, for 2 electric guitars, and rhythm section. Frog Peak Music.

— 1992. *Two Children's Songs*. Score for two tubas, tuba and trombone, or any two bass winds. Frog Peak Music.

— 1987. "Morphological Metrics: Introduction to a theory of formal distances." *Proceedings of the ICMC*. Urbana.

— 1986. *Distance Musics*. Performed at Network Muse concert. San Francisco. "Score" published in *Perspectives of New Music*, Vol. 25, No. 1 and 2. 1987.

Polansky, L., and Bassein, R. 1990. "Possible and Impossible Melody: Some Formal Aspects of Contour." Paper given at the Society for Music Theory, Oakland. 1990. Forthcoming, *Journal of Music Theory*.

Polansky, L., and McKinney, M. 1991. "Morphological Mutation Functions: Applications to Motivic Transformation and to a New Class of Time Domain Cross-Synthesis Techniques." *Proceedings of the ICMC*. Montreal.

Acknowledgements

I would like to thank Eric Smith for his invaluable contributions to the theory and implementation of the mutation functions; Martin McKinney, for ideas that found their way into this paper; Charles Ames for his expertise and invaluable advice on the clumping algorithms; and James Tenney for his assistance in the development of the clumping equation.

---

[1] Of course many magnitude interval measures are possible, including ratiometric ones. The software allows for a wide range of measures, selectable by the user. For many purposes, however, the arithmetic absolute value measure stated here works quites well.

[2] In other words, the interval is left unchanged.

[3] The *IUIM* is the "complement" of the *LCM*, "pasting" magnitude rather than sign.

[4] The *ISIM* is simply a "replacement" of values from one morphology to another.

[5] The *VM* is simply a crossfade between points of S and T, and in many cases (depending on how intervals are taken), is equivalent to the *USIM*. If intervals are taken to the value zero (external fundamental), then the *USIM* is the *VM*.

[6] In searching for a mutation such as the CCM (combinatorial contour mutation), the mutational equivalent of the ordered combinatorial direction metric (OCD) [Polansky, 1987], I was interested in a non-stochastic algorithm to generate a combinatorial contour matrix some distance between two morphologies. I used a stochastic procedure in the piece *Distance Musics* [Polansky, 1986; Polansky and Bassein]. Eric Smith and I have informally established, however, that no simple mechanism for this can exist, at least in the way I had previously defined it. A detailed explanation of this is beyond the scope of this paper.

[7] I am grateful to my longtime colleague Charles Ames for succintly clarifying this point for me, and indeed, suggesting the path to its current solution [Ames, 1991a, 1992]. My concept of clumping owes a tremendous amount to Ames' work on probability distributions and statistical feedback [Ames, 1991, 1990, 1986, and many other articles]. Ames' earlier concepts of "lumpiness" [1986] and "heterogeneity" [1990] are in fact quite similar to my "clumping," though my measure tends to work with grouped values, whereas Ames' measures tend to deal with degrees of order in a stochastically generated set [Ames, 1992].

[8] Clumping may proceed from the beginning, middle, end or any offset into a morphology. For example, maximal clumping ($\partial = 1$) with $\Omega = \mathcal{S}$, may refer to the last half of the morphology or half around the middle. The mutation software keeps this concept general.

[9] I am grateful to James Tenney for his collaboration in the development of this equation.