
The Quantization of Musical Time: A Connectionist Approach

Author(s): Peter Desain and Henkjan Honing

Source: *Computer Music Journal*, Vol. 13, No. 3 (Autumn, 1989), pp. 56-66

Published by: [MIT Press](#)

Stable URL: <http://www.jstor.org/stable/3680012>

Accessed: 02-10-2015 13:26 UTC

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at <http://www.jstor.org/page/info/about/policies/terms.jsp>

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.



MIT Press is collaborating with JSTOR to digitize, preserve and extend access to *Computer Music Journal*.

<http://www.jstor.org>

Peter Desain and Henkjan Honing

Centre for Art, Media, and Technology
Utrecht School of the Arts
Lange Veststraat 2b
NL-3511 BK Utrecht
The Netherlands

Music Department
City University
Northampton Square
London EC1V OHB
United Kingdom

The Quantization of Musical Time: A Connectionist Approach

Introduction

Musical time can be considered to be the product of two time scales: the discrete time intervals of a metrical structure and the continuous time scales of tempo changes and expressive timing (Clarke 1987a). In musical notation both kinds are present, although the notation of continuous time is less developed than that of metric time (often just a word like “rubato” or “accelerando” is notated in the score). In the experimental literature, different ways in which a musician can add continuous timing changes to the metrical score have been identified. There are systematic changes in certain rhythmic forms: for example, shortening triplets (Vos and Handel 1987) and timing differences occurring in voice leading with ensemble playing (Rasch 1979). Deliberate departures from metricality, such as rubato, seem to be used to emphasize musical structure, as exemplified in the phrase-final lengthening principle formalized by Todd (1985). In addition to these effects, which are collectively called *expressive timing*, there are nonvoluntary effects, such as random timing errors caused by the limits in the accuracy of the motor system (Shaffer 1981) and errors in mental time-keeping processes (Vorberg and Hambuch 1978). These effects are generally rather small—in the order of 10–100 msec. To make sense of most musical styles, it is necessary to separate the discrete and continuous components of musical time. We will call this process of separation *quantization*, although the term is generally used to reflect only the extraction of a metrical score from a musical performance.

Computer Music Journal, Vol. 13, No. 3, Fall 1989,
© 1989 Massachusetts Institute of Technology.

Perception of Musical Time

Human subjects, even without much musical training, can extract, memorize, and reproduce the discrete metrical structure from a performance of a simple piece of music—even when a large continuous timing component is involved. This is surprising, given that the note durations in performance can deviate by up to 50 percent from their metrical values (Povel 1977). Indeed, it seems that the perception of time intervals on a discrete scale is an obligatory, automatic process (Sternberg, Knoll, and Zukofsky 1982; Clarke 1987b). This so-called categorical perception can also be found in speech perception and vision. By contrast, the perception and reproduction of continuous time in musical performance seems to be associated with expert behavior.

Once the discrete and continuous aspects of timing have been separated by a quantization process, each can function as an input to other processes. The induction of an internal clock (Povel and Essens 1985) and the reconstruction of the hierarchical structure of rhythmical patterns (Mont-Reynaud and Goldstein 1985) both rely on the presence of a metrical score, while Todd (1985) has developed a model in which hierarchical structure is recovered from expressive timing alone.

Applications of Quantization

Apart from its importance for cognitive modeling, a good theory of quantization has technical applications. It is one of the bottlenecks in the automatic transcription of performed music, and is also important for compositions with a real-time, interac-

Fig. 1. Example of a performed score and its quantization by a commercial MIDI Package using a resolution of 1/64 note.

tive component where the computer improvises or interacts with a live performer. Last but not least, a quantization tool would make it possible to study the expressive timing of music for which no score exists, as in improvised music.



Known Methods

Few computational models are available in the literature for separating a metrical score from expressive timing in performed music (Desain and Honing 1988). Available methods produce a considerable number of errors when quantizing the data. The traditional approach is to expand and contract note durations according to a metrical grid that is more or less fixed—the grid being adjustable to incorporate different, low-level subdivisions (e.g., for triplets). Commercial MIDI software uses this method, which often gives rise to a musically absurd output, as shown in Fig. 1. Better results are obtained when the system tracks the tempo variations of the performer (Dannenberg and Mont-Reynaud 1978), though the system still returns an error rate of 30 percent. More sophisticated artificial intelligence (AI) methods use knowledge about meter (Longuet-Higgins 1987) and other aspects of musical structure. A particularly elaborate system originated at the CCRMA center at Stanford University in the automatic transcription project (Chowning et al. 1984). This knowledge-based method uses information about different kinds of accent, local context, and other musical clues to guide the search for an optimal quantized description of the data. It is entirely implemented in a symbolic, rule-based paradigm. This approach can be seen as the antithesis of our approach, in which all knowledge in the system is represented implicitly. We took the connectionist approach because knowledge-based approaches seemed to offer no real solution to manifest inadequacies of the simplistic metrical grid method. As with the majority of traditional AI programs, the sophisticated knowledge these AI methods use is extremely domain dependent (depending on a specific musical style), causing the systems to break down rapidly when applied to data foreign to this style.

Connectionist Methods

Connectionism provides the possibility for new kinds of models with characteristics traditional AI models lack, in particular robustness and flexibility (Rumelhart and McClelland 1986). Connectionist models consist of a large number of simple elements, each of which has its own activation level. These cells are interconnected in a complex network, with the connections serving to excite or inhibit other elements. One broad class of these networks, known as *interactive activation and constraint satisfaction networks*, generally converge towards an equilibrium state given some initial state.

An example of the application of these networks to music perception is given by Bharucha (1987) in the context of tonal harmony. These networks have not yet been used for quantization. The quantization model presented in this paper is a connectionist network designed to converge from nonmetrical performance data to a metrical equilibrium state. This convergence is hard wired into the system, and no learning takes place. The model is thought of as a collection of relatively abstract elements, each of which performs a rather complex function compared to standard connectionist models. While it may be possible to express these functions in terms of one of the formalisms for neural networks, this lies beyond the scope of the present article.

Basic Model

Consider a network with two kinds of cells: the *basic cell*, with an initial state equal to an inter-onset interval, and the *interaction cell*, which is connected in a bidirectional manner to two basic cells. Figure 2a shows the topology of a network for quantizing a rhythm of four beats, having its three

Fig. 2. Topology of a basic network (a) and a compound network (b).

Fig. 2

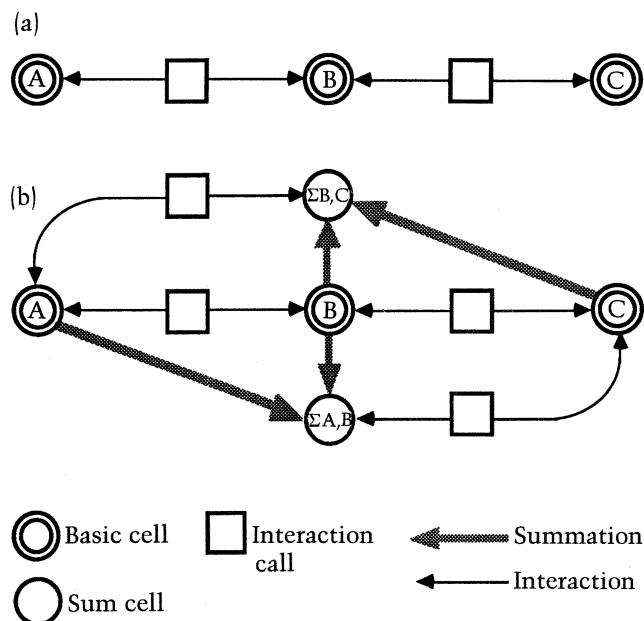


Fig. 3. Interactive time intervals in a basic network (a) and a compound network (b).

Fig. 3

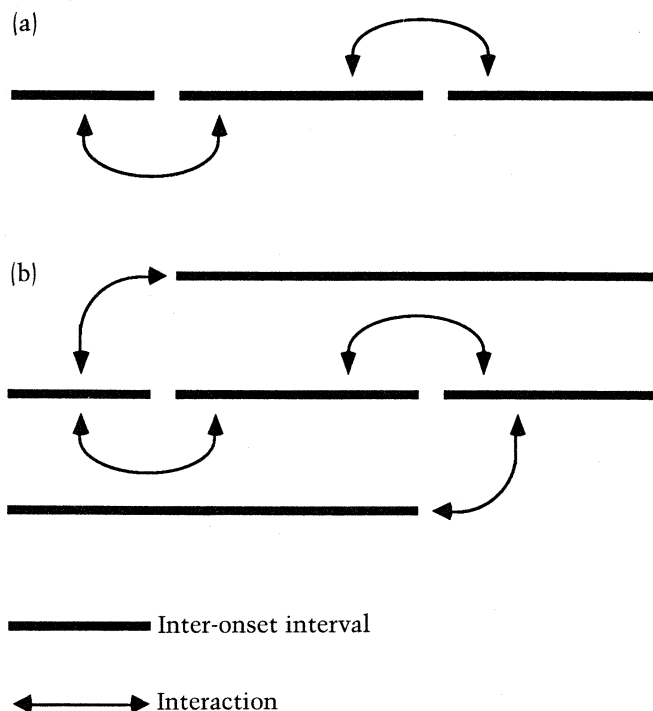
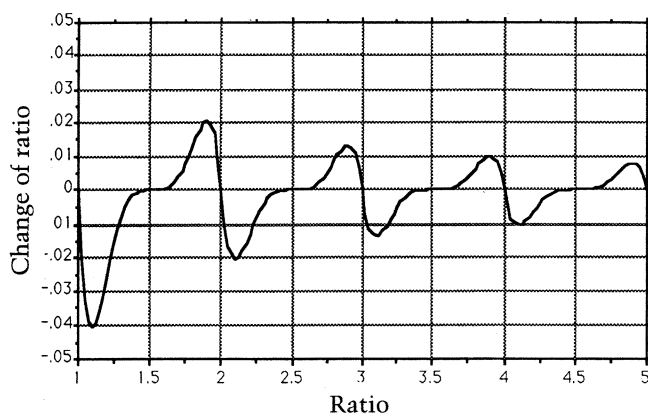


Fig. 4. Interaction function with a peak at 4 and decay equal to -1.

Fig. 4



inter-onset intervals set as states of the three basic cells, labeled A, B, and C. There are two interaction cells connected to the basic cells A and B, and B and C, respectively. Each interaction cell steers the two basic cells to which it is connected toward integer multiples of one another, but only if they are already near this state. It applies the interaction function to the quotient of their states (ratios smaller than 1 are inverted). If this ratio were close to an integer (e.g., 1.9 or 2.1), the interaction function would return a *change of ratio* that would steer the two states toward a perfect integer relation (e.g., 2). Figure 3 illustrates the interactions that are relevant in quantizing the four-beat rhythm. One can see that if the ratio is slightly above an integer, it will be adjusted downward, and vice versa as in Fig. 4.

There are constraints to be taken into account for interaction functions. First, the function and its derivative should be zero in the middle region between two integer ratios. In this region it is not clear if the integer ratio above or below is the proper goal, so no attempt is made to change the ratio. Second, the derivative around integer ratios should be negative to steer the ratio towards the integer, but greater than -1 to prevent overshoot that would result in oscillations. Third, the magnitude of the function should decrease with increasing ratios to diminish the influence of larger ratios. A large class of functions meet these constraints. At present we use a polynomial section around each integer ratio.

The degree of the polynomial, called the *peak* parameter, is typically between 2 and 12. To realize the decreasing magnitude of the interaction function, each section is scaled with a multiplication factor that is a negative power of the integer ratio. This power is called the *decay parameter*, and is typically between -1 and -3 . This interaction function is defined as

$$F(r) = (\text{round}(r) - r) * \\ |2(r - \text{entier}(r) - 0.5)|^p * \\ \text{round}(r)^d,$$

in which the first term gives the ideal change of ratio, the second term signifies the speed of change which is at maximum near an integer ratio (with peak parameter p), and the third term scales the change to be lower at higher ratios (with decay parameter d). It is simple to prove that this interaction function satisfies the constraints mentioned.

From the change of ratio $F(a/b)$, new intervals $a + \Delta$ and $b - \Delta$ are calculated without altering the sum of both intervals.

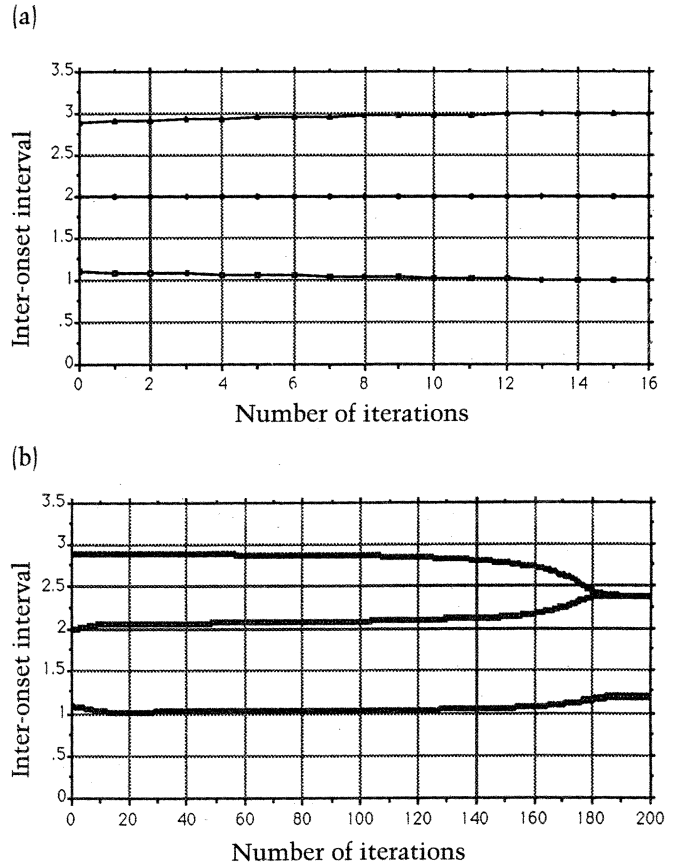
$$\frac{a + \Delta}{b - \Delta} = \frac{a}{b} + F\left(\frac{a}{b}\right)$$

which implies

$$\Delta = \frac{bF\left(\frac{a}{b}\right)}{1 + \frac{a}{b} + F\left(\frac{a}{b}\right)}$$

In simulating the network, each interaction cell updates the states of the two basic cells to which it is connected. This process is repeated, moving the basic cells slowly towards equilibrium. Equilibrium is assumed when no cell changes more than a certain amount between two iterations. For example, let us take a rhythm with inter-onset intervals of 2, 1.1, and 2.9 csec. As the representation of duration is currently unimportant in the model, they are treated as relative values (tempo has no influence on the quantization). This rhythm is represented in a basic network as three cells with the initial states 2.0 : 1.1 : 2.9. Iterating the procedure outlined above

Fig. 5. State as a function of iteration count for the rhythm 2, 1, 3 in a basic network (a). State as a function of iteration count for the rhythm 1, 2, 3 in a basic network (b).



for the interactions between cells labeled A and B, and cells B and C will adjust the durations toward 2 : 1 : 3, where the net reaches an equilibrium. Figure 5a is a graph of the state of each basic cell as a function of the iteration count.

This type of network can of course only quantize very simple rhythms. Consider for instance the rhythm 1.1 : 2.0 : 2.9, which should converge to 1 : 2 : 3. The cell representing 2.9 only interacts with its neighbor 2.0, the resultant ratio 1 : 45 being a long way from an integer. The basic net adjusts these values to 1.2 : 2.4 : 2.4, as seen in Fig. 5b.

What the model fails to take account of is the time interval 3.1, the sum of the first two durations. If this interval were incorporated into the model, it would interact successfully with the third interval (2.9) in such a way that the pair of intervals would gravitate toward the ratio 1. This observation leads to a revised model.

Fig. 6. State as a function of iteration count for a complex rhythm in a compound network.

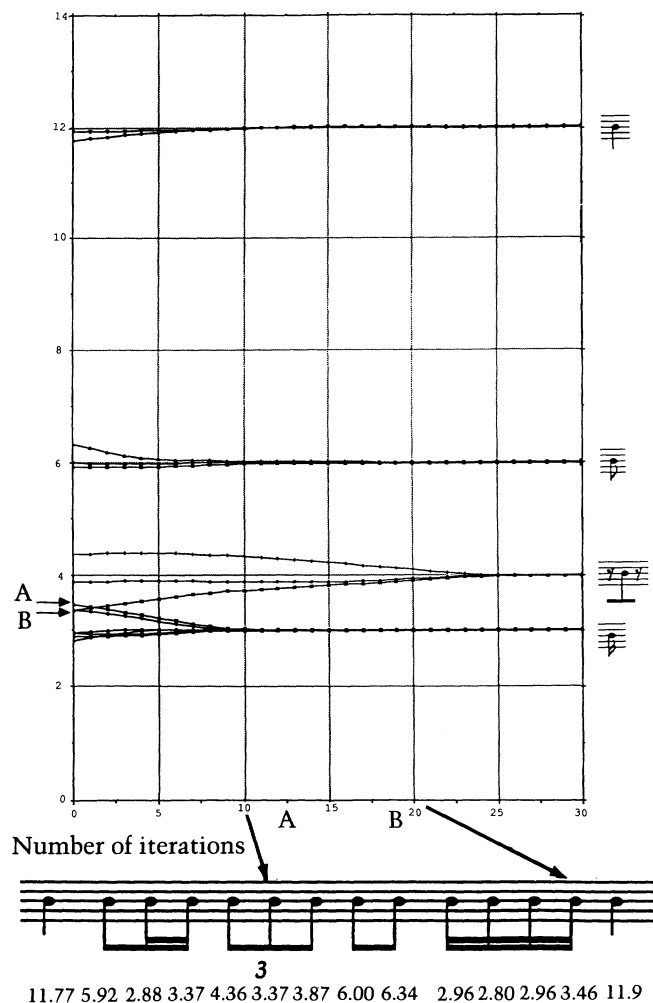
Compound Model

In order to represent the longer time intervals generated by a sequence of notes, *sum cells* are postulated. These cells sum the activation levels of the basic cells to which they are connected. The interaction of a sum cell with its basic cells is bidirectional; if the sum cell changes its value, the basic cells connected to it will all change proportionally. The sum cells are interconnected to cells representing adjacent intervals by the same interaction cells that are used in the basic model. The function of the interaction cells is once again to try to steer the interconnected cells—which may be sum cells, or a mixture of sum cells and basic cells—toward an integer ratio as was shown in Figs. 2b and 3b.

Our earlier example—a duration sequence of 1.1, 2.0, 2.9—is now quantized correctly due to combined effects of interacting sum cells and the interactions between the basic cells. Let us consider a more complex example using the real performance data shown in Fig. 6. In this rhythm the final sixteenth note is played longer than the middle note of the triplet. Nonetheless the local context of the two intervals steers each note towards its correct value as seen in Fig. 6. The compound model produces promising results, even though the network is rather sparse, allowing only adjacent time intervals to interact. A compound network for a rhythm of n intervals consists of n basic cells, $[(n + 1)(n - 2)/2]$ sum cells, and $[n(n^2 - 1)/6]$ interaction cells.

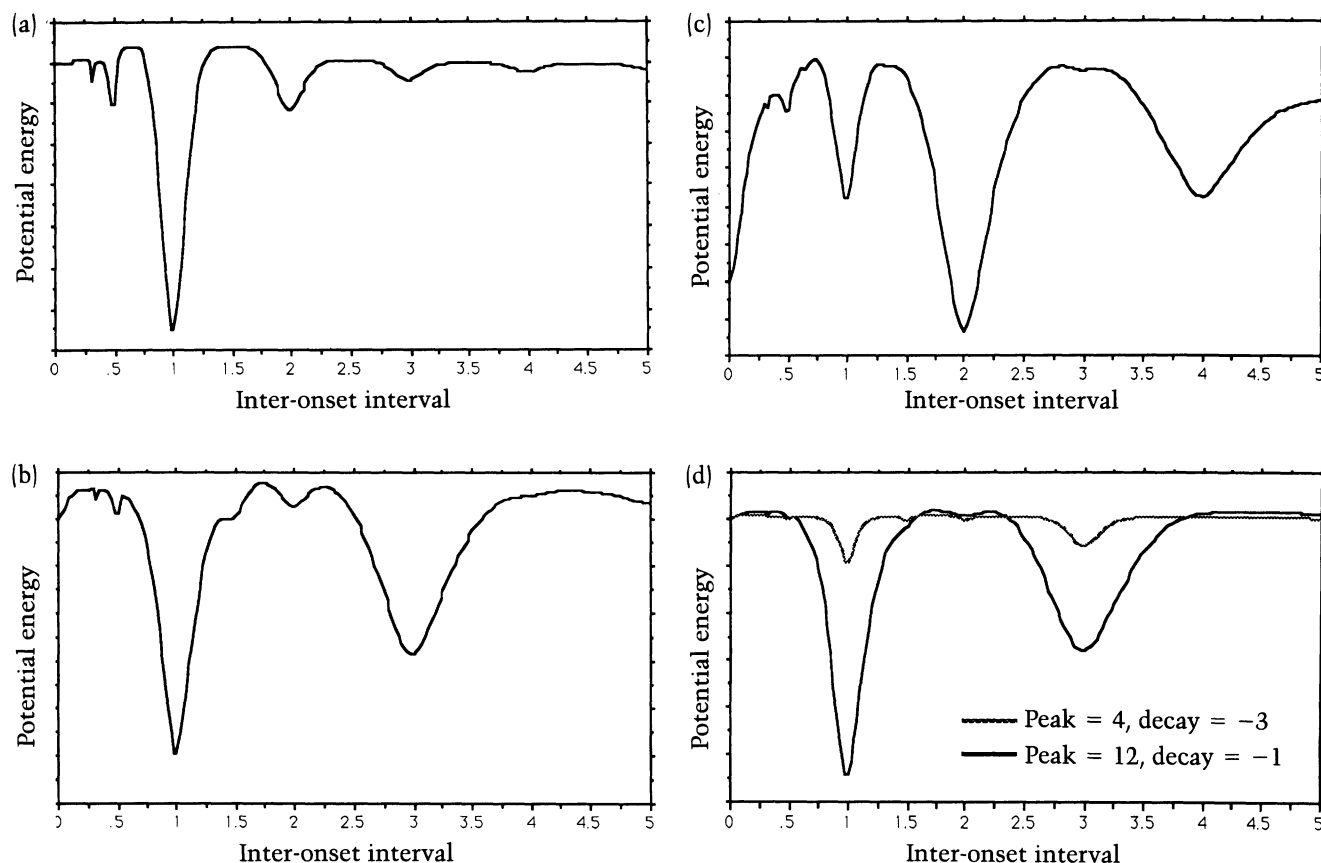
Understanding the Model

In connectionist systems the global behavior emerges from a large number of local interactions. This makes it very difficult to study the behavior of the network at a detailed level. While it may initially seem attractive to use descriptions like “winning cells,” “pulling harder,” etc., a better understanding of the patterns of change within the network and of the influence of context requires the development of specialized methods. An approach that has proved very useful is what we call the *clamping method*. This entails the clamping, or



fixing, of the states of all but one of the cells. The remaining cell is given an activation level in a reasonable range (the independent variable). Then the resulting change that would have taken place—after one iteration—if the cell were free to change its activation level is monitored (the dependent variable). In order to facilitate the interpretation of this measure (the amount of change), the function is negated and integrated to give a curve with local minima at stable points. The state of the experimentally varied cell will tend to move towards a minimum, like a rolling ball on an uneven surface. As such, it can be interpreted as a curve of potential energy. These minima and maxima can now be evaluated and judged in light of the context set up

Fig. 7. Clamping curve for a cell with a left context of 1 (a). Clamping curve for a cell with a left context of 2, 1 (b). Clamping curve for a cell with a left context of 2, 1, 1 (c). Clamping curve for a cell with a left context of 2, 1 with different parameters for the interaction function (d).



by the surrounding clamped cells. We call the interval between two neighboring local maxima the *catch range*. A value occurring within this range will move towards the minimum between these two maxima, provided the context does not change. The size of the interval where the potential energy stays close to a minimum is called its *flatness value*. It is a measure of the lack of clarity in the context; simple and clear contexts give rise to sharp minima.

Figure 7a shows the potential energy curve of two cells in a basic network; the first has a state of 1, while the other varies between 0–5. The figure shows prominent local minima at 1, 2, 3, 4 and so on, and at the inverse ratios (.5, .33, and so on). These will be the equilibrium states of the second cell. Note the flatter minima at larger ratios.

A graph of the basic interaction (without sum cells) in a 3 cell net with the first two cells clamped to the values 2 and 1 would yield the same curve,

since the first cell does not interact with the varying third cell. Introducing sum cells, however, gives a different curve as can be seen in Fig. 7b. A minimum is shown at 3 caused by the interaction of the sum of the first and second basic cells with the last cell (3:3 yielding a ratio of 1). The minimum at 3 being strengthened by the interaction of the first cell with the sum of the second two (2:4, yielding a ratio of 2). This interaction also results in a weaker minimum at 1.5 (3:1.5, a ratio of 2). With a left context of 2:1:1 the minimum at 3 almost disappears as in Fig. 7c. There is now a strong minimum at 2 because the sum cell—which combines the durations of the second and third cell—is also 2. The sum of the first three cells give rise to the minimum at 4. This clamping method thus gives a clear picture of the mechanisms involved in the complex interactions through a simplification of the process that assumes fixed values in most of the cells. The

Fig. 8. Clamping curves of two notes in the context of an idealized complex rhythm (a). Clamping curves of two notes in the context of a performed complex rhythm (b).

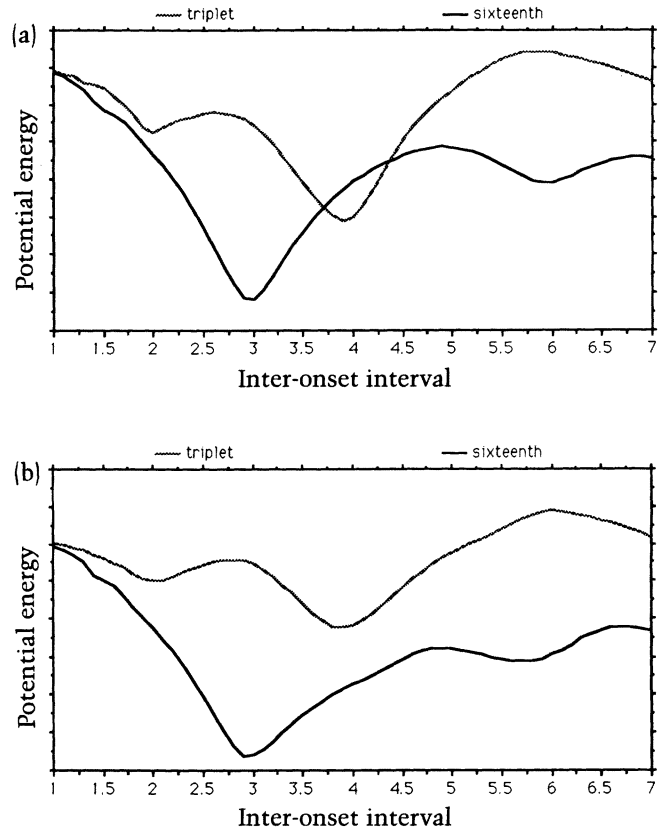
same method can also be used to study the influence of the parameters of the interaction function. In Fig. 7d, which uses the same context as in Fig. 7c, the peak and decay parameters have been changed, showing the effect on the catch range.

If we now return to the more elaborate example shown in Fig. 6, we can study the behavior of the net using the clamping method. Fig. 8a shows the potential energy curves resulting from applying the clamping method to the middle note of the triplet and the final sixteenth note. It shows clearly that the different contexts in which they appear result in different curves and that both will be directed towards the appropriate values. Note the wide catch ranges that allow rather large deviations to be quantized correctly and the smoothness of the curves. This smoothness (the lack of small local minima in the curve) is a result of the large number of interactions (364 and 91 for the triplet and sixteenth notes, respectively), which combine additively to yield each point on the curve. When the clamping experiment is rerun with performance data as context, more complex curves result, with a smaller catch range and a greater flatness, which is shown in Fig. 8b. Nonetheless, the durations still converge towards the correct metrical values.

The position of local maxima in the energy curves constitute the boundaries between the categories into which the data will be quantized. As a result, precise predictions can now be made about the perceptual interpretation of rhythmical sequences with a range of experimentally adjusted durations. It is our intention to compare these predictions with the results of empirical studies.

Implementation

In simulating a connectionist network, the calculated change in the state of one cell can be effectuated immediately (*asynchronous update*), or can be delayed, effectuating the change of all interactions at once (*synchronous update*). For asynchronous updates, a random order of visiting cells is generally preferred. In Table 1, a simplified implementation of the quantization model is given in Common Lisp (Steele 1984), based on synchronous updates.



The basic cells are represented as a vector of inter-onset intervals. The sum cells are not represented explicitly, but are recalculated, summing the represented interval of basic cells for each interaction. A macro is provided that implements the iteration over adjacent sum intervals. The described interaction function is the one we used for the Figs. 5 and 6. This simplified version requires the minimum inter-onset interval to be around 1. More elaborate versions run in Common Lisp and in C on stock hardware (Macintosh II and Atari ST series machines).

Further Research

The model we have presented needs high peak values to stabilize accurately. Because this results in smaller catch ranges, we are currently studying the automatic increasing of the peak parameter while

Table 1. Micro version of the connectionist quantizer in CommonLISP

```

;;; MICRO CONNECTIONIST QUANTIZER
;;; 1988 P.Desain and H.Honing

;;; Utilities

(defmacro for ((var &key (from 0) to) &body body)
  "Iterate body with var bound to successive values"
  (let ((to-var (gensym)))
    '(let ((,var ,from) (,to-var ,to))
      (loop , (when to '(when (> ,var ,to-var) (return)))
        ,@body
        (incf ,var))))))

(defmacro max-index (vector)
  "Return index of last element in a vector"
  '(- (array-dimension ,vector 0) 1))

(defmacro zero-vector! (vector)
  "Set elements of a vector to zero"
  '(for (index :from 0 :to (max-index ,vector))
    (setf (aref ,vector index) 0.0)))

(defmacro incf-vector-scalar! (a b from to)
  "Increment elements in a range of a vector"
  '(for (index :from ,from :to ,to)
    (incf (aref ,a index) ,b)))

(defmacro incf-relative-vector-vector! (a b)
  "Increment elements of a vector proportionally"
  '(for (index :from 0 :to (max-index ,a))
    (incf (aref ,a index) (* (aref ,a index) (aref ,b index)))))

(defun print-vector (times vector &optional (stream t))
  "Print all elements of vector"
  (format stream "~%~3d: " times)
  (for (index :from 0 :to (max-index vector))
    (format stream "~2,1,5$ " (float (aref vector index)))))

;;; control structure for iteration over intervals

(defmacro with-all-intervals (vector (begin end sum) (start
finish) &body body)
  "Iterating over all intervals contained in [start,finish]"
  '(let (,sum)
    (for (,begin :from ,start :to ,finish)
      (setf ,sum 0.0)
      (for (,end :from ,begin :to ,finish)
        (incf ,sum (aref ,vector ,end))
        ,@body))))

```

(cont'd)

```

(defmacro with-intervals (vector (begin end sum) (start
finish) &body body)
  "Iterating over intervals"
  '(let ((,sum 0.0)(,begin ,start))
    (for (,end :from ,start :to ,finish)
      (incf ,sum (aref ,vector ,end))
      ,@body)))

(defmacro with-adjacent-intervals
  (vector (a-begin a-end b-begin b-end a-sum
b-sum) &body body)
  "Iterating over interval pairs"
  '(let ((max-index (max-index ,vector)))
    (with-all-intervals ,vector (,a-begin ,a-end
,a-sum) (0 (1- max-index))
      (with-intervals ,vector (,b-begin ,b-end ,b-sum)
        ((1+ ,a-end) max-index)
        ,@body))))

;;; Main quantization procedures

(defun quantize! (durations &optional (peak 4)(decay -1))
  "Quantize data in durations vector"
  (let ((changes (make-array (length durations) :initial-
element 0.0)))
    (for (times :from 0)
      (print-vector times durations)
      (update! durations changes peak decay))))

(defun update! (durations changes peak decay)
  "Update all durations synchronously"
  (zero-vector! changes)
  (with-adjacent-intervals durations
    (a-begin a-end b-begin b-end a-sum b-sum)
    (let ((delta (if (> a-sum b-sum)
                      (delta (/ a-sum b-sum) peak decay)
                      (- (delta (/ b-sum a-sum) peak decay)))))
      (incf-vector-scalar! changes (/ delta a-sum) a-begin
a-end)
      (incf-vector-scalar! changes (- (/ delta b-sum) b-begin
b-end)))
    (incf-relative-vector-vector! durations changes)))

(defun delta (ratio peak decay)
  "Return change of time interval"
  (let ((delta-ratio (interaction ratio peak decay)))
    (/ delta-ratio (+ 1 ratio delta-ratio))))

```

(cont'd)

```

(defun interaction (ratio peak decay)
  "Return change of ratio"
  (let ((position (1- (* 2 (- ratio (floor ratio))))))
    (goal (round ratio)))
    (* (- goal ratio)
      (abs (expt position peak))
      (expt goal decay))))

;;; usage examples
;;; minimum element in data should be larger than 1

(quantize! (vector 1.1 2.0 2.9))
(quantize! (vector 11.77 5.92 2.88 3.37 4.36 3.37 3.87
                  6.00 6.34 2.96 2.80 2.96 3.46 11.93))

```

the network comes to rest. The dependency of the model on absolute time and absolute tempi is still an open question. The most difficult rhythmic cases for this model are: (1) those that involve additive durations that emerge when rests and tied notes occur in the data and (2) divisive rhythms, such as when a quintuplet is adjacent to a triplet. Our aim is to be able to characterize exactly the limits of the model and to evaluate the computational requirements and the psychological plausibility of the results. A further aim is to develop a robust technical tool for real-time quantization using a process model. Tempo tracking is then an absolute necessity.

Conclusion

We consider the compound model presented here to be promising. In difficult cases the system undergoes a graceful degradation instead of a sudden breakdown: that is, the range in which rhythms are caught and quantized correctly becomes more and more limited. However, it is a paradoxical problem with connectionist models that their adaptability means that even a rough first implementation, with obvious bugs, may exhibit appropriate behavior. In order to increase an understanding of the process involved, it is necessary to develop specialized tools for diagnosis and investigation. The clamping

method described here seems to have considerable potential, and we are confident that further tools of a similar sort will develop as connectionist modeling gathers momentum.

Acknowledgments

We would like to thank Dirk-Jan Povel, Steve McAdams, Marco Stroppa, the reviewers of *Computer Music Journal*, and especially Eric Clarke and Klaus de Rijk for their help in this research and their comments on the first version of this paper.

References

- Bharucha, J. J. 1987. "Music Cognition and Perceptual Facilitation: A Connectionist Framework." *Music Perception* 5(1): 1–30.
- Chowning, J., et al. 1984. "Intelligent Systems for the Analysis of Digitized Acoustical Signals." *CCRMA Report* STAN-M-15.
- Clarke, E. 1987a. "Levels of Structure in the Organization of Musical Time." *Contemporary Music Review* 2: 212–238.
- Clarke, E. 1987b. "Categorical Rhythm Perception: An Ecological Perspective." In A. Gabrielsson, ed. *Action and Perception in Rhythm and Music*. Stockholm: Royal Swedish Academy of Music. No. 55: 19–33.
- Dannenberg, R. B., and B. Mont-Reynaud. 1987. "An On-

- line Algorithm for Real Time Accompaniment." *Proceedings of the 1987 International Computer Music Conference*. San Francisco, California: Computer Music Association, pp. 241–248.
- Desain, P., and H. Honing. 1988. "The Quantization Problem: Traditional and Connectionist Approaches." *Proceedings of the first Artificial Intelligence and Music Workshop*. St. Augustin, West Germany: Gesellschaft für Mathematik und Datenverarbeitung.
- Longuet-Higgins, H. C. 1987. *Mental Processes*. Cambridge, Massachusetts: MIT Press.
- Mont-Reynaud, B., and M. Goldstein. 1985. "On Finding Rhythmic Patterns in Musical Lines." *Proceedings of the 1985 International Computer Music Conference*. San Francisco, California: Computer Music Association, pp. 391–397.
- Povel, D. J. 1977. "Temporal Structure of Performed Music: Some Preliminary Observations." *Acta Psychologica* 41:309–320.
- Povel, D. J., and P. Essens. 1985. "Perception of Temporal Patterns." *Music Perception* 2(4):411–440.
- Rasch, R. A. 1979. "Synchronization in Performed Ensemble Music." *Acustica* 43(2):121–131.
- Rumelhart, D., and J. McClelland, eds. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, Massachusetts: MIT Press.
- Shaffer, L. H. 1981. "Performances of Chopin, Bach, Bartok: Studies in Motor Programming." *Cognitive Psychology* 13:326–376.
- Steele, G. L. 1984. *Common Lisp: The Language*. Bedford, Massachusetts: Digital Press.
- Sternberg, S., R. L. Knoll, and P. Zukofsky. 1982. "Timing by Skilled Musicians." In D. Deutsch, ed. *The Psychology of Music*. New York: Academic Press.
- Todd, N. P. 1985. "A Model of Expressive Timing in Tonal Music." *Music Perception* 3(1):33–58.
- Vorberg, D. J., and R. Hambuch. 1987. "On the Temporal Control of Rhythmic Performance." In J. Requin, ed. *Attention and Performance VII*.
- Vos, P., and S. Handel. 1987. "Playing Triplets: Facts and Preferences." In A. Gabrielsson, ed. *Action and Perception in Rhythm and Music*. Royal Swedish Academy of Music. No. 55:35–47.