Contents

1 Introduction **6**

2 Aspects of Music Psychology **11**

2.1 Introduction 11

2.2 What are the perceived elements of melody? 11

2.2.1 Review of experiments 11

2.2.2 Summary 23

2.3 The Perceptual Grouping of Musical Elements 24

2.3.1 Scope 24

2.3.2 Primitive Grouping Mechanisms 25

2.3.3 The Parsing of Music and Other Serial Patterns 26

2.3.4 Summary 38

2.4 Expectation in Music Psychology 38

2.4.1 Review 38

2.4.2 Summary 52

3 Neural Networks and Music **54**

3.1 Introduction 54

3.2 Neural network applications in music 56

3.2.1 General review 56

3.2.2 Models of memory for musical sequences 60

3.3 Adaptive Resonance Theory and its application 69

3.3.1 Introduction 69

3.3.2 The **ART2** network 69

3.3.3 **ART2** Simulation 75

3.3.4 **ART2** performance with a chord classification task 76
3.3.5 L' ART pour l'art 82
3.4 Miscellaneous 84
3.4.1 More Applications of Neural Networks in Music 84
3.4.2 Models of Sequential Memory 85

4 The Masking Field and the MASKART network 88
4.1 Introduction 88
4.2 Representing Temporal Order with Spatial Patterns 88
4.3 The Temporal Chunking Problem 91
4.4 The Masking Field 93
4.5 Computer Simulations of a Masking Field 95
4.6 The results of the initial simulations 99
4.7 The **ART3** algorithm 108
4.8 Extending the masking field 111
4.9 Constraints on the operation of a masking field within **ART3** 118
4.9.1 "Input" and "Mismatch" Reset 118
4.9.2 **ART3** dynamics 123
4.9.3 Learning should be slowed when expectations are active 123
4.9.4 The equilibrium size of bottom-up weight vectors 124
4.9.5 Resets 125
4.10 Constraints on the operation of $\mathcal{F}_0$ 126
4.10.1 Limiting the capacity of working memory 126
4.10.2 The "banana problem" 127
4.11 The setting of parameters 128
4.12 Summary 129

5 Application of the MASKART network 130
5.1 Experiments showing the effect of learning on masking field classifications and expectations 130
5.2 Running **MASKART** with musical sequences as input 137
5.2.1 The input data set 137

5.2.2 The evolution of working-memory patterns . . . 138
5.3 Experiments presenting **MASKART** with individual melodies . . . 143
5.3.1 Procedure . . . 143
5.3.2 Overview of errors . . . 155
5.3.3 A second experiment with altered input . . . 156
5.3.4 Errors . . . 171
5.3.5 Presentation of the complete (altered) input set . . . 171
5.3.6 Experiments with equilibrium weights . . . 176
5.4 Review of Gjerdingen (1992) . . . 179
5.5 Summary . . . 184
5.5.1 Problems with the current formulation . . . 184

6 SONNET 1 **189**
6.1 Introduction . . . 189
6.2 **SONNET 1** . . . 189
6.2.1 **SONNET 1** is initially homogeneous . . . 190
6.2.2 The calculation of $\mathcal{F}_2$ input . . . 190
6.2.3 The learning of bottom-up LTM traces . . . 191
6.2.4 Specification of the set T_i . . . 193
6.2.5 The evolution of $\mathcal{F}_2$ activations . . . 194
6.2.6 The learning of $\mathcal{F}_2$ lateral inhibitory weights . . . 195
6.2.7 Configuring $\mathcal{F}_1$ as a dynamic working memory . . . 196
6.2.8 The addition of feedback weights . . . 197
6.3 Timing Issues . . . 198
6.4 Simulating SONNET 1 . . . 200
6.5 **SONNET 1** simulations . . . 203
6.5.1 Introduction . . . 203
6.5.2 The setting of parameters . . . 205
6.5.3 Modifications to the **SONNET 1** network . . . 206
6.5.4 Incremental learning of sequences . . . 208
6.5.5 Learning with "cyclic" presentation . . . 210
6.6 Modifying the conditions for top-down feedback . . . 213

6.6.1 The calculation of W^+ and $W^\times$ . . . 222
6.6.2 "Chunking out" . . . 224
6.7 Summary . . . 224

7 Extensions to SONNET 1 **225**
7.1 Introduction . . . 225
7.2 Dealing with repeated items . . . 226
7.2.1 Proposed solution . . . 226
7.2.2 Implementation . . . 228
7.2.3 Results . . . 232
7.2.4 The generation of expectations . . . 240
7.3 The provision of a hierarchy . . . 241
7.3.1 Structure . . . 241
7.3.2 Implementation . . . 245
7.3.3 Learning phrases of phrases . . . 247
7.3.4 Hierarchical expectations . . . 250
7.4 Summary . . . 265

8 Modelling aspects of rhythmic structure **267**
8.1 Introduction . . . 267
8.2 The need for a rhythmic pulse generator in SONNET . . . 267
8.3 The gated pacemaker model . . . 270
8.4 Synchronizing a pacemaker with a rhythmic stimulus . . . 273
8.4.1 Formalizing the problem . . . 273
8.4.2 The phase-locked loop . . . 276
8.5 A neural phase-locked loop . . . 277
8.6 Initial results . . . 280
8.7 Tempo range, spontaneous tempo and preferred tempo . . . 288
8.8 Coping with "missing" onsets . . . 291
8.9 Alternative pacemakers . . . 294
8.10 The work of Jones et. al . . . 299
8.11 Tempo-invariant durational encoding . . . 301

8.11.1 Grossberg's "rhythm generator" 302
8.11.2 An alternative scheme 303
8.11.3 The chunking of durational patterns 308
8.11.4 Reading a melody out of long-term memory 310
8.12 Summary 312

9 Conclusions **313**
9.1 Summary 313
9.2 Further work 316

Chapter 1

Introduction

Music has been a focus of intellectual activity throughout history. The Greek thinkers, notably Pythagoras, developed a system of thought that united music, mathematics, religion and cosmology. The discovery that the degree of consonance of musical intervals was related to the simplicity of the ratio of the corresponding tone-frequencies was momentous, in that it was one of the first examples of "the mathematization of human experience" ([94]). The influence of the Greeks continued throughout the Middle Ages, through the work of Boethius, Aurelian, Regino etc., and is evident in Kepler's *Harmony of the World* (1618). Influenced by the Cartesian scientific method, Mersenne, Saveur and Rameau developed the science of acoustics, and in the nineteenth century Helmholtz's *Sensations of Tone* was not only of scientific importance but also prompted the design of new instruments and the development of new compositional techniques.

There also emerged competing music theories, which attempted to explain "the way music worked" in a pseudo-scientific way. Indeed, Forte [58] describes the work of Schenker, one of the foremost of this century's music theorists, as being able to "be likened to a particular kind of high-level achievement in science: the discovery or development of a fundamental principle which then opens the way for the disclosure of further new relationships, new meanings". Schenker's work details how music (in the western tonal idiom) can be analyzed by a process of reduction, representing a kind of inverse of the compositional technique of variation, to reveal how the musical surface (the *foreground*), constitutes, via intermediate structural levels (the *middleground*), an expression of a fundamental progression, the *Urlinie*, which resides at the *background* level and which exerts almost mystical control over the entire work. Forte

emphasizes that this analytical edifice was not based on "abstruse speculation, or in acoustical or metaphysical formulations ... but ... [on] aural experiences with actual musical compositions", but nevertheless Schenker's theory, though derived from the listening experience, does not seek to give an account of that experience.

More recent work, for instance that of Meyer [109][110], places more emphasis on the impact that music has on the mind of the listener. Meyer is particularly concerned with the way in which music can convey meaning, other than simply referential meaning, and can evoke an emotional response. Central to his theory is the concept of *expectation* (later termed *implication*): his basic hypothesis is that "affect or emotion-felt is aroused when an expectation ... activated by the musical stimulus situation, is temporarily inhibited or permanently blocked". He is somewhat critical of the disciples of Schenker for placing too much emphasis on the highest structural levels whilst minimizing the rôle played by the dynamic interplay of expectations and realizations, at levels closer to the musical surface, in conveying both affect and embodied meaning. The concept of a hierarchy of structural levels persists, however, and is indeed ubiquitous in theories of music. Meyer is less clear in his specification of the source of musical implications — he makes the obvious point that expectations must be at least partially based on the "habit responses developed in connection with particular musical styles", but also invokes aspects of Gestalt Psychology, in particular the "laws" which apparently underlie a large variety of perceptual phenomena and which are therefore often held to be innate.

Jackendoff [81] highlights one issue which appears difficult for Meyer's theory of emotion in music, namely the fact that pieces of music which are sufficiently familiar for there to be no doubt about the identity of the forthcoming events, can continue to elicit an affective response — in other words, affect persists even though expectations are fully realized. Jackendoff in turn postulates the existence of a musical "parser" that operates independently of specific (veridical) memories. He refers to Lerdahl and Jackendoff's "Generative Theory of Tonal Music" [105], itself heavily influenced by Chomsky's work on language, and proposes it as a model for the operation of the musical parser. The grammar proposed by Lerdahl and Jackendoff is "intended as as an account of the *experienced* listener's 'final-state' understanding of pieces" [my italics], highlighting that, although the parser is conceived as operating independently of specific memories, it is nonetheless a function of the listener's experience with, in this case,

the western tonal idiom. Whilst Lerdahl and Jackendoff avoid any temptation to speculate how the principles on which the generative theory is based might be instantiated in the brain, they place their theory very much in the domain of cognitive science. They summarize by positing that "many of the principles of musical grammar are not specific to music ... that musical grammar is a confluence of a number of factors independently needed for more essential aspects of cognitive ability ... [and] that it can provide central evidence toward a more organic theory of mind".

Narmour [115][116] seeks to build a comprehensive model of implication and realization, with particular reference to melodic structure. His theory makes distinct the contributions of two expectation systems: a top-down system, that is "flexible, variable and empirically driven"; and a bottom-up "brute force " system that is unconscious and automatic. He bases his detailed exposition of the latter system on the Gestalt laws of similarity, proximity and common fate, as applied to the registral and intervallic aspects of melodic motion. Top-down "style-based" implications are able to suppress those of the bottom-up system, but not to the point where the latter disappear — this is consistent with Jackendoff's belief that conflicts between automatic and learned expectations are capable of generating affect even in familiar contexts. Interestingly, Narmour's bottom-up system is not entirely divorced from learning issues, since, although he postulates that the basic characteristics of *process* and *reversal* are innate, he admits that the "material" on which they act is necessarily culturally acquired. Once again, Narmour's theory emphasizes the cognitive aspects of music theory.

Smoliar [145] criticizes Narmour's description by questioning some of the assumptions on which it is based: for instance, he questions whether melody can *necessarily* be isolated from the other components of heard (as opposed to notated) music; secondly, whether music necessarily has a syntax metaphorically related to that of natural language; and thirdly, whether notes are necessarily the lexical primitives of such a syntax. Smoliar suggests that another approach, that of perceptual categorization, where "categories are *formed* in the course of perception", is more appropriate to the analysis of music as perception. He cites the work of Edelman [52], who has hypothesized that "the only 'universal' capacity that sensing animals have is the capacity to develop [such] archetypes in response to sensory experiences". Smoliar's view is consistent with the approach taken in this thesis, whereby the perception of, in this case , melodies is modelled with a system that learns by exposure to a musical

"environment". Smoliar sees this view of perceptual organization as being antithetical to the approach expounded by Narmour — their disagreement is reminiscent of the long-running nature/nurture debate.

It is apparent, even from this quick review, how music theory has become intimately concerned with subject areas traditionally considered the domain of cognitive scientists and psychologists. This movement has been accompanied by a growing interest in the "psychology of music". Some of the work in this area, particularly as it relates to the perception of melodies, is reviewed in Chapter 2.

The study of psychology, cognitive psychology in particular, has become increasingly influenced by progress in the field of "artificial intelligence", i.e. the computer modelling of intelligent behaviour. Much of the work in this field has been concerned with the rule-based processing of "symbolic" knowledge by dedicated "expert systems". In recent years, however, there has been a revival in interest in so-called "connectionist" models, which employ (virtual) networks of simple processing units to perform tasks at a "sub-symbolic" level. The structure of these "neural networks" is intended to be analogous to that of the brain, though this analogy is often rather tenuous. Chapter 3 reviews applications of connectionism to the study of music perception and, more generally, to the learning of sequences. A critique of certain "supervised" models of sequence learning is given. A class of self-organizing, unsupervised networks is introduced and the relative appropriateness of such networks is discussed with regard to music perception. An example is given of how such networks can be employed in a chord-classification task, and the effect of manipulating various of the network parameters is illustrated.

Chapter 4 concentrates on the development of a self-organizing network for the learning of sequences. The "masking field" model is introduced and the results of computer simulations of its behaviour are presented. An extended masking field is then incorporated into a larger ART-type network, **MASKART**, which is applied, in Chapter 5, to a melody-learning task. **MASKART**'s drawbacks are discussed in detail and possible remedies are suggested.

Chapter 6 introduces the **SONNET 1** network: the first independent simulations of that network are described and various modifications are made. The application of **SONNET 1** to sequence learning is investigated and its advantages over the **MASKART** model are discussed. A detailed study of priming within the **SONNET 1** network is described.

Chapter 7 extends the **SONNET 1** network to enable it to handle item repetitions and hierarchical structure. The design of a computer simulation of the extended network is described. The extended network is then applied to the melody-learning task and its performance of melody recognition and recall is assessed. This chapter contains the first application of a **SONNET**-type network to a real-world problem.

Chapter 8 details a connectionist approach to the modelling of aspects of rhythm perception. The means by which a small network of simple units can "track" a rhythmic stimulus is addressed, and the resulting framework is applied to the problem of tempo-invariant durational encoding. The relevance of the "rhythm-network" to the models introduced in earlier chapters is discussed, and the design of a combined network for the perception of rhythmic patterns is suggested.

Chapter 9 summarizes the achievements of this work and suggests promising avenues for further research.

Chapter 2

Aspects of Music Psychology

2.1 Introduction

In this chapter I will deal with issues concerning the ways in which humans *perceive* music, how these ways are dependent on the limitations of the brain as information processor and how they are influenced by an individual's musical enculturation. In short, this chapter will review aspects of the psychology of music. Of course, "the psychology of music" refers to an enormous body of work, so this chapter will necessarily cover in detail only those issues which pertain directly to the models developed in the later chapters, whilst only touching upon others less immediately relevant. Nevertheless, I believe that a knowledge of such issues is both essential for the development of realistic models and useful in their subsequent assessment. I will begin by considering the elements out of which a mental representation of melody is formed.

2.2 What are the perceived elements of melody?

2.2.1 Review of experiments

The problem of determining the "building blocks" out of which a listener constructs a mental representation of a simple melody has generated much interest among psychologists and they have devised many experiments to test the appropriateness of various types of description: possibilities include the representation of a melody as an ordered set of absolute pitches, as an ordered set of relative pitches, as an ordered set of intervals or as a contour (the sequence of "ups and downs"). This section contains a review of some of these experiments and the

conclusions drawn from them.

White [156] took a set of 10 well-known melodies and produced "distorted" versions of them by performing 12 transformations on the interval sequences corresponding to the melodies. All 10 melodies were subjected to the same set of transformations. Some of the transformations were linear, e.g. each interval was doubled, whilst some were non-linear e.g. each interval was set equal to one semitone whilst maintaining the sign. The 130 resultant melodies (10 straight plus 120 distorted) were then played to Subjects whose task was to identify the original melody from the distorted version in a forced choice between the ten named melodies. This therefore was a test of Subjects' ability to access melodies stored in Long Term Memory (LTM). The results indicate that Subjects were able to identify the melodies at levels above chance for all the transformations, including the transformation setting all intervals to zero ("$i = 0$"). This particular result indicates that Subjects were able to use duration alone to aid in melody identification. White noted that the transformations which least impaired recognizability were those which "left the relative sizes of intervals unchanged as well as the sequences of ups and downs", whereas transformations which changed either of these properties had a much more pronounced effect. Interestingly, the transformation which maintained the set of intervals but randomized their temporal ordering, led to better recognizability than either the "$i = 0$" transformation or the transformation decreasing all intervals by 2 semitones (with the proviso that intervals were not allowed to change sign). White concluded that this result lent support "to the supposition that S's depended heavily on interval-information in identifying melodic patterns".

Deutsch [35] attempts to summarize the operations that are necessary for music recognition: abstraction of intervals (both those formed by simultaneous and successive notes); abstraction of chords consisting of at least three notes; and the identification together of tones separated by octaves. Drawing some inspiration from the studies into visual abstraction by Hubel and Wiesel [78], she suggests a two-channel mechanism capable of performing these abstractions (see Figure 2.1). Channel A first classifies specific intervals and chords (e.g. the major second between C and D) and then abstracted intervals and chords (e.g. any major second). In the case of intervals the distinction between simultaneous and successive is made as above. Channel B first maps frequency-specific neurons onto a single octave array and then maps the elements of this array in groups of three onto "third-order neurons". These

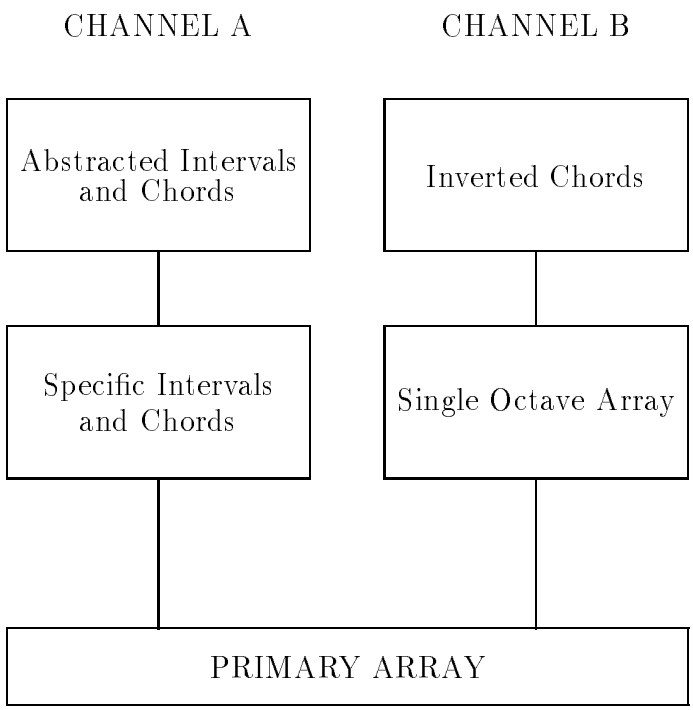


Figure 2.1: Deutsch's Two Channel Network for Music Recognition (after Deutsch (1969))

third-order neurons thus classify chords irrespective of their inversion. Deutsch briefly reviews some neurophysiological evidence for the existence of auditory neurons with complex responses, though she is careful not to use these as proof for the existence of neurons performing the particular operations detailed above. She is also neutral regarding the issue of whether the abstractions are learned or innate, although she cites Allen [2], who showed that Subjects with musical training rated tones separated by an octave as more similar than did untrained Subjects.

Dowling and Fujitani [48] point out the importance of contour information as a unifying influence within pieces of music of many different kinds. They emphasize that such unification is achieved in spite of the destruction of exact interval sizes and the possibility of an intervening change of key. Furthermore they support the conclusion of White [156] that "preserving the relative sizes of successive intervals seems important in the development of the phrase". They then describe two experiments: the first is intended to model the short-term recognition of

"subject-answer pairs" in, for instance, a fugue; the second tested the recognition of familiar folktunes distorted in one of three ways: so as to preserve the contour and relative sizes of successive intervals; so as to preserve only the contour; or so as to preserve the first note of each measure (with other notes changed "to destroy contour but preserve the implicit harmonic basis of the melodies"). As in White's experiment Subjects were given a set of possible responses, in the form of named tunes, from which they were forced to make a choice on each trial. They differentiate the latter experiment from White's experiment firstly by using tunes with identical rhythmic patterns and secondly by ensuring that when they generated the distorted versions they selected intervals, when required, from a set of intervals with a distribution approximating the distribution found in folktunes (and indeed western music in general).

In the first experiment Subjects were divided into three groups: one group heard a standard melody (which differed for each trial) and after a 2 second pause heard either an exact copy of the first melody or a random collection of notes; a second group heard the standard melody followed by either the same melody again or a melody with contour preserved; a third group heard the standard melody followed by either a melody preserving contour or a random collection of notes. The task of the first two groups was to judge whether the comparison melody was identical to the standard or not. The task of the third group was to judge whether the two melodies had the same contour. For all the groups, half of the Subjects performed the task with the comparison melody beginning on the same note as the standard, whilst half had "transposed" melodies. The results of the first experiment indicate a slight difference in strategies for recognition between the transposed and untransposed conditions. For the untransposed condition Subjects apparently used the recognition of specific pitches to aid in comparison. In the transposed condition, however, the Subjects appeared to rely heavily on contour, easily distinguishing the same and same-contour conditions from random melodies but finding it very difficult to distinguish between the same and same-contour conditions themselves. They note that short-term recognition did not seem to depend on the perception of exact interval sizes.

The results of the second experiment indicated that Subjects were slightly better able to identify well-known tunes from distortions that preserved contour and relative size of successive intervals as opposed to distortions preserving contour alone. The level of performance

under either of these conditions was substantially worse than that for undistorted tunes. Dowling and Fujitani conclude that information concerning exact interval sizes was much more important in the recognition of familiar melodies (presumably stored in LTM) than in the first experiment's testing of Short Term Memory (STM). They suggest that "memory for exact interval sizes may depend on extensive learning of specific melodies, and thus the effect would appear in long-term memory for well learned melodies but not in the short-term memory conditions of Experiment 1". This distinction between the factors that influence short-term recognition and those that influence long-term recognition is an important one, which should be borne in mind when assessing the results of this type of experiment.

Dowling [42] performed an experiment to test whether the melodic transformations of inversion, retrograde and retrograde inversion, which can all be found in western music, are actually perceivable by the listener or if they are just the "intellectual conceits of the composer". He then used the results to assess the plausibility of various models of the psychological processes at work. The experiment itself tested the recognition of the three transforms listed above, under conditions which varied the speed of presentation (fast or slow) and whether exact-interval size or just contour was preserved during the transformations. The results indicated that all three transformations could be recognized at levels greater than chance. The ascending order of difficulty was inversion, retrograde and retrograde inversion. Low presentation speeds enhanced recognition but there was no significant difference between the exact-interval and contour transforms. The fact that retrograde inversions were the most difficult to recognize leads Dowling to prefer the "pitch-vector" characterization of the process by which Subjects handle the task, over the "interval-vector" characterization i.e. he favours a mechanism that represents a melody as a sequence of pitches relative to an arbitrary reference rather than as a sequence of intervals. This preference is based on relating the difficulty of a task to the number of "operations" required to perform the task but Dowling admits the tentative nature of his conclusion by suggesting other possible explanations for the data. Neither does he seek to choose between two suggested strategies for the performance of the task, one in which Subjects manipulate the pitch-vectors themselves and the other in which Subjects manipulate images of pitches, e.g. imagined notes on a page.

Idson and Massaro [80] perform four experiments to assess the applicability of a bidimensional model of pitch in melody recognition. They summarize the work of Deutsch [35][36],

who concluded that melodic perception was mediated by perception of successive intervals and that this process could be disrupted by randomly distributing the notes of a melody between three octaves whilst maintaining "tone chroma". Idson and Massaro suggest that Deutsch was premature in ruling out chroma information as an important factor in melody recognition and hypothesize that the disruption described above was caused by the fact that Deutsch did not preserve the contour of the melodies when displacing the notes by octaves. They point out that had both contour and tone chroma been preserved then the so-called "abstract intervals" would have been maintained, in spite of the loss of tone height and absolute interval information. The first experiment therefore tested the recognition of five well-known melodies under various transformations: OVC (mnemonic: **O**ctave-displaced, **V**iolated **C**ontour) replicated Deutsch's (1972) transformation by displacing tones by octave intervals and violating contour; OPC displaced tones by octaves but preserved contour; PC preserved contour only; LT (Linear Transform) halved the size of intervals, thus preserving contour and the relative size of successive intervals. The recognition performance after each of these transforms was compared with the performance for the untransformed tunes and the decrement in each case gave an indication of the importance to melodic recognition of the information that had been lost during that transform. It was found that performance for OPC was not significantly different from performance with the untransformed tunes (causing only a 3 percent decrement), whereas transform OVC caused a 40 percent decrement. According to the authors, these results indicate that chroma information was indeed important as long as contour was preserved. The enhanced recognition for OPC was shown not to be due to contour alone by the fact that both LT and PC gave a decrement in performance of over 20 percent. Experiments 2, 3 and 4 disconfirm possible alternative explanations for the results of the first experiment. These alternatives concerned respectively the possible enhanced discriminability of contour when large intervals are present, the possible usefulness of a contour preservation that is highly accurate in some sense and the possibility that contour solely contributes to the identification of abstract interval magnitude, irrespective of direction. Idson and Massaro thus summarize that a bidimensional model of pitch is applicable to melodic perception and that tone chroma, when combined with contour preservation, can give valuable information for melody recognition.

Idson and Massaro's interpretation of these results, namely that using tone chroma without

contour makes the individual interval classes ambiguous and that this is the explanation for the Deutsch's results, is rejected by Deutsch herself. She argues that the results of Idson and Massaro might have been achieved by a process of "hypothesis testing" i.e. that Subjects, being presented with a short list of possible responses in the form of melody names, might be able to hypothesize the correct melody name from the contour cue alone and then to subsequently confirm or disconfirm this hypothesis by comparing the tone chroma of the expected notes with that of actual notes as they arrive. In this manner, the preservation of contour might lead to improved performance without the need to postulate the existence of interval-class detectors. Such an explanation would be consistent with her model of music recognition [36]. Kallman and Massaro [89] attempt to exclude this possible explanation by re-running the Idson and Massaro experiment with a slightly different procedure. In this experiment, therefore, Subjects were asked to identify melodies in the same way as before but they were not given a range of possible answers, nor were they presented with a particular tune more than once. Performance, in terms of the percentage of correct identifications (averaged over three presented tunes) was as follows: in condition O (original tunes) performance was 92%; in condition OPC (random octaves, contour preserved) performance was 61%; in condition OVC (violated contour) performance was only 3%. To distinguish what part of this effect was due to contour preservation alone, a second experiment was performed with an extra condition, viz. PCS, which preserved contour but slightly stretched it so as to destroy chroma information. In this experiment mean performance for OPC was 72%, for OVC it was 8% and for PCS it was 11%. It is important to note the figure for PCS is not significantly greater than the figure for OVC. This result, if repeated for a larger set of data, would seem to run counter to Deutsch's hypothesis testing theory—if Subjects can hypothesize the identity of a tune from condition OVC to a level not significantly worse than in the PCS condition, as this result appears to imply, then they ought to be able to use "hypothesis testing" in the OVC condition also. This would not then provide an explanation for the results of Idson and Massaro's original experiment or indeed Kallman and Massaro's first experiment. In this regard it should be noted that Deutsch's assertion that contour alone is a sufficient cue with which to hypothesize melody identity rests on data from White [156] and Dowling and Fujitani [48] in both of which experiments Subjects were given a small (10 member and 5 member respectively) set of possible familiar melodies from which to choose. It is at least possible that contour is a

viable cue to hypothesis generation only in this type of forced choice paradigm (which would include Idson and Massaro [80]) and not in the unlimited choice paradigm of Kallman and Massaro [89]. If this is the case then Deutsch's hypothesis testing strategy should not operate in Kallman and Massaro's experiment and the 64% increase in performance between conditions OVC and OPC would appear to be only explainable in terms of chroma/interval class. Deutsch [39], however, points out that in Idson and Massaro's experiment there is only a 3% decrement in performance between conditions O and OPC whereas in Kallman and Massaro there is a mean 25% difference between these conditions. This difference is hard to explain on the assumption that retention of pitch class and contour is sufficient to identify a melody, no matter what the set of possible responses.

It is hard to make any definite conclusions from this series of experiments, particularly since between them they employ different procedures (some used a forced-choice paradigm others an unlimited choice) and in some cases only small amounts of data were generated. In addition, there may be parallel recognition processes at work, each utilizing different information to identify a melody.

Dowling [44] develops a two-component model of melody storage in both long- and short-term memory. The first component is the contour and the second is the musical scale "on which the ups and downs of the contour were hung". He hypothesizes that "contour is an abstraction from the actual melody that can be remembered independently of the pitches or interval sizes" and that this applies to both long- and short-term memory. Dowling reviews the finding of Dowling and Fujitani [48] that, in the transposed condition, exact-same targets and same-contour targets were almost impossible to tell apart but notes that the fact that that the target melodies were atonal, and thus lacking a familiar musical scale, was crucial to this finding. He further predicts that "distinguishing between a tonal melody and a tonal lure having the same contour but starting on a different note of the same modal scale should be very difficult" because they share both mode and "diatonic interval pattern". This prediction was borne out by an experiment in which subjects were asked to say whether the second melody of a pair was the same as the first (a tonal melody), in the sense that it was an exact transposition. It was found that two melodies were easy to distinguish if they had different contours and that if the second melody preserved contour but was atonal then Subjects found it easier to distinguish than did the Subjects in Dowling and Fujitani (1971), who were presented with two

atonal melodies. Most strikingly, it was found that Subjects only performed at chance levels when asked to distinguish between exact transpositions and tonal lures as defined above. Dowling claims that this result demonstrates that the functions of contour and mode are separate—as he puts it "the mode is simply a framework on which the contour may be hung". As far as long-term memory is concerned, Dowling cites Attneave and Olson [5], who show that for familiar melodies exact interval sizes are remembered. This suggests that a model of long-term memory for melodies should utilize more than just contour and mode. He goes on to propose a parsimonious way of coding melodies in long-term memory which relies on the fact that small intervals predominate in all world musics. It should be noted, however, that this scheme for "quasi-linguistic marking" is devised with the intention of reducing the "load on long term memory" when compared with a model in which every interval size is remembered. It appears that this supposed reduction of information is heavily contingent on the way in which interval events are represented—if intervals are coded by the activation of their corresponding neurons (as in [35]), it is difficult to see how Dowling's method of coding results in any reduction in memory load.

Dowling [45] tries to establish the ways in which Subjects of varying levels of musical training represent melodies in short-term memory. He quickly rules out representation as a set of absolute pitches (as do all the studies listed here, either implicitly or explicitly) since such an encoding strategy would only be available to those with "perfect pitch". The three other possible strategies he considers are representation by melodic contour, by intervals or by relative pitches (by which he means a set of scale steps, e.g. "do","re","mi" etc.). He points out that the usefulness of the latter would be limited by the Subject's ability to assign a tonal framework to the piece (though other studies have shown this task to be rapidly accomplished, where appropriate, by those familiar with traditional western music). In the first of three experiments he presented Subjects with brief novel melodies and then tested them for recognition after a filled 40-second interval. Subjects were asked to distinguish between exact transpositions of the target melodies and imitations of the targets in which one of the notes had been changed. To establish a tonal framework, and thus make possible a scale-step representation of the melody, each of the melodies was surrounded by a context consisting of a chord progression strongly suggestive of a particular tonal framework. Thus by varying the tonal framework for a particular melody and, say, its exact comparison (taking

into account the fact that all the comparison stimuli were transposed from the original), the scale step representation of the melody would change whilst the interval and contour representations would remain the same. The experiment was carried out using Subjects from three groups: musically inexperienced, moderately experienced and music professionals. Dowling found that the inexperienced and moderately experienced Subjects performed "better than chance and about equally well overall ..., whereas professionals performed better". Interestingly, however, the inexperienced Subjects performed equally well regardless of tonal context whilst the moderately experienced Subjects "performed better than inexperienced listeners with same-context items; their performance fell to chance when context changed ...". The performance of the professionals was approximately equal for both conditions.

Dowling concludes from these results that inexperienced listeners appear to use an interval-based strategy, moderately experienced listeners appear to use a scale-step representation whilst professionals seem to be able to adapt their strategy to the task in hand.

The second experiment ruled out an alternative explanation for these results based on the details of the experimental procedure whilst the third experiment attempts to establish whether the scale-step representation consists of a direct encoding of scale-step or rather a "contour-plus-tonal-framework" encoding. The results show that moderately experienced listeners were better able to recall contour information when required, a necessary condition for the latter means of encoding (Dowling's preferred option). It should be noted that this finding does not preclude the former option and Dowling falls back on "prior evidence", including that of Deutsch [37], on which to base his preference. In doing so, he appears to confuse tone chroma, in the sense that Deutsch refers to it, with scale-step. I believe that this is an error, albeit a subtle one, since the concept of tone chroma does not include the notion of a tonal framework, i.e. all occurrences of the note A♯ fall into the A♯ chroma regardless of their tone height, but only within the tonal context of the key of A♯ will the note A♯ constitute the tonic scale-step. Therefore it may well be that listeners with a certain amount of musical experience can encode melodies via scale-step representations, but that this was obscured in experiments such as in [36] by the fact that listeners were unable to establish a tonal framework. This in turn may have been due to the unusually large intervals which prevail in this type of experiment. Much has been written about the "extraction" of tonal hierarchies (Krumhansl and Shepard [99], Castellano, Bharucha and Krumhansl [26] etc.) and it is not my intention

to review this body of work here. However, I would point out that both Bharucha [8] and Deutsch [40] state that a "native" listener may make use of information from the temporal ordering of tones to divine rapidly the tonal hierarchy of piece. By contrast, the non-native listener has to rely on a longer-term assessment of the tonal "statistics" in order to achieve the same result. It is thus at least plausible that the unusual multi-octave melodies in the Deutsch experiment [36] may have precluded the establishment of a firm tonal context, thus rendering the task of melody identification surprisingly difficult, even for professionals who might otherwise have been able to use scale-step representations to aid in the task.

Dowling and Bartlett [47] performed a set of experiments which are important in establishing the differential influence of certain musical parameters in short- and long-term recognition tasks. In two experiments Subjects were played a melodic figure taken from a Beethoven piece and then, after a "filled" five minute interval, they were presented with another figure which was one of three types: "Targets" were exact copies of the original figure; "Relateds" shared the contour of the original figure but not exact interval sizes; "Lures" were figures from the same piece which shared neither contour nor exact intervals. The task was to differentiate both Targets and Relateds from Lures, and further, if possible, to differentiate Targets from Relateds. The results of these experiments surprised the authors, indicating that Targets were much better recognized than Relateds, recognition of the latter being only slightly above chance levels. These results suggested that Subjects were relying on exact interval information for recognition rather than on contour information, in spite of the fact that, during the initial presentation phase, Subjects were asked to perform an additional task which was intended by the authors to draw attention to the contour of the figures.

Dowling and Bartlett hypothesized that this result, unexpected as it was in the light of previous research detailing the short-term effects of contour preservation (e.g. [44]), may have been due to their choice of real musical stimuli as opposed to the artificially generated melodic figures used in previous work. To test this hypothesis they performed two further experiments using such artificial stimuli and again the results led them to the conclusion that "contour information is easily extracted from novel musical stimuli, but contributes to performance only with short (and/or unfilled) intervals ...[whereas] interval information is difficult to extract, but contributes ... to performance over a broad range of retention intervals".

To account for these results the authors suggest a possible mechanism involving two sep-

arate memory "traces": a short-term trace representing absolute pitch, contour and key, and a long-term trace representing ordered interval/chroma information. They propose that a stimulus may only "access" a trace by matching information explicitly represented in that trace e.g. a long-term memory trace might only be accessed by a stimulus containing the correct intervals in the correct order. Once accessed, however, "rehearsal" of this trace could generate a short-term trace containing contour information for comparison purposes. This explanation fits well with the findings of Kallman and Massaro [89], in which contour by itself appeared to be ineffective in accessing long-term memories corresponding to familiar melodies. These conclusions are supported and refined by subsequent experiments by Edworthy [53] and Dowling [46].

Edworthy [53] carried out six short-term recognition experiments to ascertain conditions under which Subjects make use of contour information as opposed to interval information. She concluded that contour information was available to Subjects in the short-term regardless of novelty or transposition but that its usefulness decayed with the length of the stimuli. By contrast, interval information was imprecise in cases where Subjects found it difficult to identify key (e.g just after transposition to a distant key) but was immediately and precisely available once key had been firmly established. In these circumstances, interval information was resistant to decay in memory, particularly in cases where the stimulus melodies were familiar to Subjects. These observations suggest that interval information might be stored in long-term memory by representing a melody as a series of scale-notes (as suggested in [45])—this would explain why the accuracy of interval information would depend on the firm establishment of a key.

Dowling [46] also expresses support for the notion that contour dominates recognition for non-familiar melodies in the short-term but that after longer delays the relative importance of exact interval information increases. Dowling performed an experiment designed to test memory for contour and exact interval sizes over both short unfilled and long filled delays. He used target melodies of differing "tonal strengths" in an attempt to ascertain the effect that this parameter had on melodic memory. The results of this experiment led Dowling to two main conclusions: firstly that false alarms to same-contour-different-interval lures decreased over the longer delay period and secondly that tonal strength "was an important determinant of this effect". This is compatible with the hypothesis that exact-interval representations are

favoured in strongly tonal contexts. In addition, Dowling concedes that the fact that false alarm rates to new contour lures rose with weakening tonality suggests that tonality affects memory for contour. This contradicts [44], in which he postulates that contour and scale are encoded independently. As he puts it "Both here and in Bartlett and Dowling's study [6], listeners did not ignore interval information in encoding contour, and the contours of melodies that matched tonal scale invariants of the culture were best remembered".

2.2.2 Summary

Melodies are not remembered as ordered sets of absolute pitches. This is obvious from the universal ability to recognize a melody irrespective of the absolute pitch of its initial note. If we exclude those individuals possessing "perfect pitch", memory for absolute pitches can be considered to be a short-term phenomenon. The body of experimental work cited above suggests that information regarding the contour of a melody is likewise only directly available in the short-term. Its "usefulness" in the kind of tasks described is maximal when no "better" information is available, e.g. when the sense of tonality is weak.

Long-term memory for melodies is mediated by a mechanism which takes into account exact interval sizes and/or relative pitches (scale-notes). It is difficult to choose between these two possible descriptions and there is some evidence that both are available to Subjects to a degree dependent on their musical experience. It is thus possible that, as children, we represent melodies as sequences of intervals but that, with an growing awareness of a tonal framework, the salience of the scale-note representation is correspondingly increased. In the melody identification tasks involving octave-scrambled melodies, contour information was shown to affect the levels of performance. This effect was most drastic in forced-choice tasks, where at least part of the effect could be ascribed to the usefulness of contour information in confirming/disconfirming hypotheses. However, the apparent effectiveness of the preservation of contour information in increasing performance in unlimited-choice tasks, in which chroma information was also preserved, suggests that the availability of contour information in some way disambiguates the interval/scale-note information, which in turn drives the recognition process. In terms of an interval representation this implies the existence of "abstract-interval detectors", each responding to a set intervals of the same registral direction, differing in magnitude by a (small) integral number of octaves, e.g. a major second and a major ninth. In

terms of a scale-note representation this implies a perceptual difference between, for instance, rising to the tonic and falling to the tonic; this suggests that the range of a set of scale-note detectors might extend beyond the single octave most usually assumed.

In the absence of a clear-cut advantage for either the interval or the scale-note representation, I chose to use the representation most suited to the aims of this thesis. Throughout the thesis I am concerned with modelling the perception of sequences, particularly tonal melodies in the western idiom. In this regard, notions of expectation and parsing, and the way in which they relate to long-term memory, become important. Such concepts are best dealt with in an explicitly tonal context: as Dowling puts it in [45] " ... the scale steps of melodic pitches are inseparable from their tonal functions, which in the diatonic tonal system carry their meanings ...". In tonal contexts a single tone can elicit learned expectations whilst the expectations arising from an interval can differ depending on the scale-notes that bracket the interval. Thus I shall prefer the scale-note representation of melodic sequences in long-term memory, whilst bearing in mind the probable parallel existence of LTM for interval sequences.

2.3 The Perceptual Grouping of Musical Elements

2.3.1 Scope

Having examined the various possible representations of musical events, I will now consider the ways in which these events are grouped together during perception. It will first be necessary to restrict the scope of such an investigation to that area with which this thesis is primarily concerned, namely the effect that *learned schemata* have on the processing of musical sequences. The next section will therefore only briefly summarize those aspects of auditory event grouping which are believed to rely on "primitive" mechanisms, that is mechanisms which are innate rather than learned. It should be noted, by way of a caveat, that the distinction which is made between these two types of mechanism is essentially the same distinction as underlies the "nature-nurture" debate which has been evident throughout much of the history of psychology/artificial intelligence. The extent of this debate is testament to the difficulty of drawing a firm line to demarcate the precise contributions of innate processes/learned knowledge to a given behaviour and perhaps suggests that one should treat any such line as somewhat arbitrary.

2.3.2 Primitive Grouping Mechanisms

Bregman [18] reviews in great detail the topic of "auditory scene analysis", i.e. the ways in which the brain is able to analyze the mixture of sounds with which it is commonly presented in such a way that the organism can perceive as grouped those sounds which can plausibly be supposed to emanate from the same event or source. Bregman proposes that this analysis is accomplished in two ways: "...by the use of primitive processes of auditory grouping and by governing the listening process by [learned] schemas that incorporate our knowledge of familiar sounds ...". He points out that such grouping has to be carried out in both the spectral and temporal dimensions and adopts the terms "sequential integration" and "simultaneous integration" to emphasize this distinction. Primitive sequential integration is seen to depend on factors including frequency proximity, temporal proximity, spatial location, timbre and modulation, and manifests itself most often as "auditory stream segregation", whereby the auditory input apparently divides into separate "streams". During streaming the spectral and temporal relationships between events in different streams, and therefore any patterns which these events might form, are difficult to perceive. Primitive simultaneous integration also depends on the factors listed above as well as on, for instance, the relatability of various partials present in the sound to a common fundamental frequency. Bregman shows how these various processes are exploited in western music e.g. in the creation of compound melodic lines in Baroque music by the same instrument rapidly alternating between separate pitch ranges, and also shows how composers attempt to avoid inappropriate grouping by the use of, for instance, dissonance. Huron [79] has analyzed polyphonic works by Bach to show how simultaneous intervals were carefully chosen to promote "consonance" without promoting "fusion", allowing the horizontal (melodic) aspects of such music to be more easily perceived.

The contribution of primitive processes to the perception of music is thus great. To account for the way in which we perceive music in a familiar style, however, it is also necessary to introduce the concept of learned schemata. Such schemata encapsulate information regarding regularities in the particular environment to which an organism is exposed. As Bregman puts it: "...schemas become active when they detect, in the incoming sense data, the particular pattern they deal with. Because many of the patterns schemas look for extend over time, when part of the evidence is present and the schema is activated, it can prepare the perceptual process for the remainder of the pattern". One of the aims of this thesis is to investigate

mechanisms by which schemata appropriate to the perception of western music can be learned and employed.

2.3.3 The Parsing of Music and Other Serial Patterns

I now wish to consider the way in which we parse streams of musical events into sub-groups or phrases. Issues of parsing are intimately related to those concerning the immediate memory of extended sequences. Miller [111] establishes that the span of immediate memory is of the order of seven items or *chunks* regardless of the number of bits of information contained in each chunk. Immediate recall of sequences longer than seven chunks can be achieved by *recoding* the sequence into another sequence that contains fewer chunks with more bits of information per chunk. For accurate recall this recoding must be more or less automatic and involves the use of representations stored in long-term memory. This section reviews work describing the chunking or parsing of item sequences, in particular melodies, and the effect that this has on the learning and recall of such patterns.

One of the most influential papers in the area of the parsing of item sequences is Restle [130], in which a theory of the hierarchical representation of serial patterns, not necessarily musical patterns, is developed. The theory was devised with reference to a series of experiments in which Subjects were equipped with a "response box" having six buttons in a row. Above each button there was an "event light". The lights were turned on, one at a time, in a sequence having a fixed and regular pattern, and the task of the Subject was to anticipate correctly the next light in the sequence by pressing the corresponding button. The experimental procedure consisted of varying the sequential pattern of the lights to ascertain at which locations in the various sequences the most errors in anticipation were made. Restle and Brown [132] had already excluded five conventional "stimulus-response" type theories of serial learning as candidates for an adequate theory of serial pattern learning. Restle begins by stating, with reference to both language and music, that "it seems overwhelmingly obvious that long and complex serial patterns are divided into natural subparts, and that mastery is facilitated if the incoming sequence of events is somehow marked off into natural subparts". He asks the question "...what is the nature of a 'natural' or 'useful' subpart?".

Restle's first set of experiments showed the importance of "runs of events", such as (2,3,4,5), as prominent subparts. As Restle puts it "if such a run appears anywhere in a

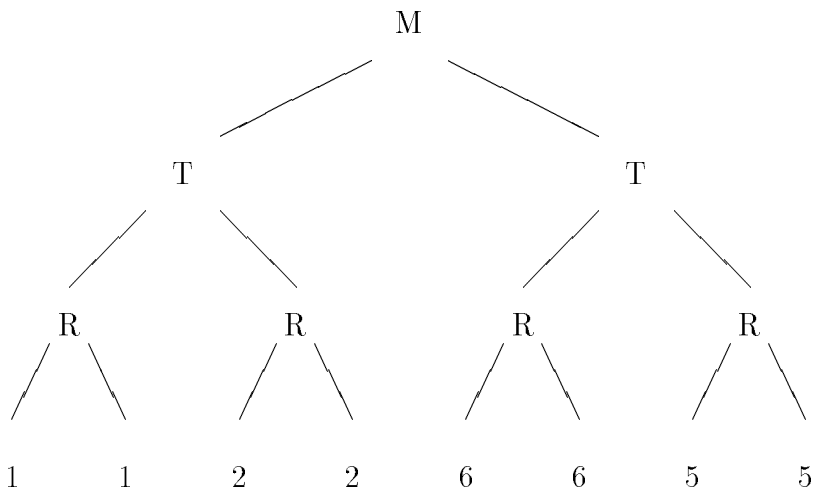


Figure 2.2: Tree diagram of sequence (1,1,2,2,6,6,5,5)

sequence, Subjects are likely to notice and use it both as a marker indicating the length of the periodic sequence and as the basis for correctly predicting the events later in the run". In addition to runs, there is also a weaker tendency to perceive "trills", such as (2,3,2,3), as salient sub-sequences. Restle postulates that runs and trills are favoured because the patterns they represent are easily generated by the application of simple rules—in the case of the run, the rule might be to turn on the light immediately to the right of the previously illuminated light. He describes Event-Interval theory (E-I theory) and its extension Recursive E-I theory before generalizing these approaches into a "theory of structural trees". This theory is at the core of Restle's approach. Its basic assumption is that certain patterned sequences are able to be generated by simple hierarchical trees (see Figure 2.2). At each node of a tree will be found "elements of a rule system". These elements can either be "rule systems"themselves or, in the case of terminal nodes, specific events. For Restle's tree structures the application of a "rule system" consists of the execution of an operation denoted by a capital letter e.g. the letter T denotes transposition by one step. In general, these operations are defined such that they subsume the concatenation operation (denoted "+") i.e. $T(X) = X + t(X)$, where t is the simple transposition operation such that $t(1,2,3) = (2,3,4)$. In this way

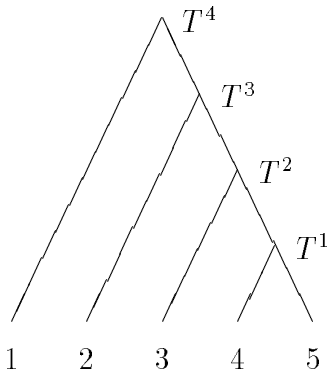


Figure 2.3: Right Branching Tree representing $T^4(1)$ (after Restle (1970))

$T(1,2,3) = (1,2,3,2,3,4)$. Restle also defines the operations $R(X)$ (repetition) and $M(X)$ (mirror), such that $R(1,2,3) = (1,2,3,1,2,3)$ and $M(1,2,3) = (1,2,3,6,5,4)$. Furthermore, structural tree theory requires that operations can act on the results of previous operations e.g. $M(T(1,2,3)) = (1,2,3,2,3,4,6,5,4,5,4,3)$. Restle claims that, in this way, long sequences can be parsimoniously encoded, and subsequently regenerated from this encoding, provided that they possess a structure amenable to such a hierarchical description. He postulates that sequences that can be encoded thus will be easier for Subjects to learn and he gives results of an experiment in which this was shown to be the case. He also shows that Subjects' error profiles are consistent with the hypothesis that "Subjects should have the most difficulty in learning and correctly predicting the highest order transformations in the sequence".

Restle and Brown [132] performed further experiments with "ambiguous" patterns i.e patterns which have two or more possible (incomplete) tree representations, for example $(2,1,2,3,4,3,4,5,6,5)$, which can be encoded $(2,1,2)(3,4,3,4)(5,6,5)$ as trills, or (2)-$(1,2,3,4)(3,4,5,6)(5)$ as runs. Interestingly, the preferred parsing was shown to depend on pretraining. Further problems are raised with regard to patterns such as $(1,1,2,2,3,3,4,4)$. This pattern is easy to learn but it is difficult to encode as a binary tree. By contrast the pattern $(1,1,2,2,2,2,3,3)$ is easily encoded as $T(T(R(1)))$, but apparently more difficult to learn. To account for this observation, Restle introduces the notion of "right-branching trees" (see Figure 2.3), which are a further generalization of the binary trees discussed above. A right-branching tree is defined recursively thus: at every level the tree divides into two parts; the left part consists of a single element or a smaller tree built on an element; the right part

is a subtree with the same property as the parent tree. This definition is summarized in the "rewrite rule" $F^n(S) \Rightarrow S + F^{n-1}(f(S))$, where S is any sequence to which f can be applied. For example

$$T^3(2) = 2 + T^2(3) = 2 + 3 + T(4) = 2 + 3 + 4 + 5$$

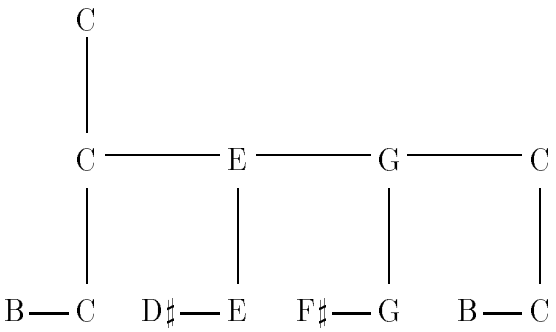
In summarizing , Restle foreshadows Deutsch and Feroe [41] by subjecting one of Bach's two-part Inventions to a structural tree analysis, though he emphasizes that this analysis is necessarily an over-simplification and intended, therefore, as an illustration of the possibilities of applying serial pattern learning techniques to the analysis of (primarily) melody.

Deutsch and Feroe [41] build on the work of Restle and others (e.g. Simon and Sumner [143]) to construct a model of what they call "the internal representation of pitch sequences in tonal music". The formal rules are complex: the following description is a distillation of those listed in the paper.

1. a *structure* consists of an ordered string of operators together with an asterisk representing a *reference element*. Possible operators include n (next), p (previous), and s (same). Thus *structure* A might be $(A_1, A_2, \ldots, *, \ldots, A_7, A_8)$, where A_n represents an operator. A *structure*, when associated with an *alphabet*, α, defines a *sequence*—this is denoted $\{A;\alpha\}$. When a *reference element*, r, is specified a *sequence of notes* is defined—this is denoted $\{\{A;\alpha\};r\}$.

2. an operator acts on the result of the operation closest to it but on the same side as the asterisk. If an operator is adjacent to the asterisk, the result is determined by that operator acting on the *reference element*. The resulting *sequence of notes* is defined as the ordered sequence of the results of individual operations.

3. A *sequence of notes* can be generated from two other *sequences* $\{A;\alpha_1\}$ and $\{B;\alpha_2\}$ by using the "prime" sequence operator [pr]. The *sequence of notes* A[pr]$B;r$ is generated by first generating the *sequence of notes* $\{A;\alpha_1\};r$ and then using each resulting note in turn as the *reference note*, r', to produce a *sequence of sub-sequences*, each sub-sequence being of the type $\{B;\alpha_2\};r'$. Thus if the *sequence* A has m elements and the *sequence* B has n elements then the compound *sequence* will have $m \times n$ elements, divided into m *sub-sequences*. Deutsch and Feroe also define the sequence operators [ret] (retrograde)



$$A = \{(*, 3n); C_{tr}\}$$

$$B = \{(p, *); Cr\}$$

$$S = \{A[pr]B; c\}$$

Figure 2.4: An Example of a sequence encoding (after Deutsch and Feroe (1981))

and [inv] (inversion), which behave in similar ways. These particular sequence operators were chosen "from considerations of musical analysis".

4. *alphabets* are ordered sets of notes. Typical *alphabets* are the notes making up a particular diatonic scale, the notes making up a given major triad etc. e.g. the *alphabet* C_{tr} would represent the set of notes {C, E, G}.

5. more complex encodings allowing a sequence to include multiple *alphabets* and allowing strings of operators to act on strings of *sequences* are also described.

An example of the type of encoding generated by the application of these rules is shown in Figure 2.4. Deutsch and Feroe claim that this type of representation has advantages over that proposed by Restle [130]: certain structural notes are represented at multiple levels and the part they play in forming higher level sequences serves "to cement the lower-level sequences together"; distinct alphabets can be associated with different levels of the hierarchy; strings of operators can be associated in chunks of (usually) three or four items, which is consistent with other research into serial recall of strings.

The rôle of alphabets in this system is an important one. The authors consider the chromatic scale as the parent of all alphabets. They further claim that the generic descriptions

of other common alphabets are "retained in long-term memory" in relation to the parent e.g. a generic ascending major scale might be stored as $\{(*, 2n^2, n, 3n^2, n)Cr\}$—one could generate any major scale simply by adding a reference note. The ease of this operation mirrors the ease with which we perceive sequences of notes in a transposition invariant manner. In this way, alphabets appear to be more like an archetypal sequence of intervals retained in long-term memory. Moreover, it is made clear that these archetypes are learned by exposure to a particular environment.

Chord progressions are also cited as examples of sequences amenable to hierarchical encoding. It is noted that chords themselves are abstractions which can result in a "many-to-one" mapping—a chord can be abstracted from any of its inversions. A chord progression is thus a learned abstraction of abstractions, which the authors liken to the use, in linguistics, of concepts such as "noun phrases". The authors postulate that "the generation of a sequence of chords might occur in parallel with the generation of a sequence of notes".

There follows a section in which Deutsch and Feroe discuss the processes by which the listener establishes a hierarchical representation from the sound stream that s/he hears, in which Deutsch and Feroe assume that initial groupings are formed on the basis of "simple perceptual mechanisms". For instance, they state that there is a strong tendency to group together elements proximal in pitch, with the proviso that the inter-group pitch difference is not so great as to prevent the key elements in each group from forming a higher-level sequence. There is an important point to make here: it is not clear that pitch-proximal groups are preferentially formed because of a primitive or innate grouping mechanism that favours such groups or, alternatively, because pitch proximal groups are learned archetypes i.e. groups of notes proximal in pitch are "chunked" by virtue of the "familiarity" of these groups. I believe the latter mechanism to be important in the perception of music in familiar styles and even more so in the perception of familiar pieces of music. Much of this thesis is devoted to issues concerning the learning of veridical archetypes (often called schemata), and the rôle that such learning has on subsequent music perception. It is very likely that both grouping mechanisms (innate and learned-schematic) are operational in parallel. In this regard, a visuo-linguistic analogy may be instructive. The grouping of the string "theCATsatONtheMAT" is facilitated both by learned schemata (words) and by a primitive grouping mechanism related to intra-group similarity and inter-group difference in letter size. This approach to parsing, which

acknowledges the importance of learning, is given support by Deutsch and Feroe both implicitly, by their deriving sub-sequences of notes from alphabets which are themselves culturally acquired, and explicitly when they say "a sequence representation must also involve complex processes, in which the listener draws on his expectations about a given musical style". The issue of learned expectations will be dealt with in the next section.

Of overwhelming importance in the parsing of sequences is the temporal dimension. Relative temporal proximity can mask the effects of other cues to grouping. Using a visuo-linguistic analogy again, the semantic grouping in the phrase "asti tchi ntim esav esni ne" is overwhelmed by, in this case, the spatial grouping. A number of studies have sought to highlight the importance of temporal grouping and the way in which it interacts with other grouping mechanisms.

Restle [131], followed up his original paper on serial pattern learning with a paper relating to the rôle of "phrasing" in the perception of such patterns. He defined "good phrasing" as the placing of relatively long pauses before the relatively important structural points in a sequence (as defined by the structural tree corresponding to the sequence). Similarly, a sequence was considered "badly phrased" if its pause structure was incompatible with its tree structure. A sequence with equal pauses between all successive notes was deemed to be "unphrased". Subjects were asked to perform a pattern learning experiment, in which they were able to preview, under various phrasing conditions, the patterns to be learnt. The patterns were presented at one of two speeds ("allegro" and "adagio") in an attempt to ascertain whether the relative or the absolute pause times were significant determinants of subsequent performance. The results show that well-phrased previews led to significantly better learning than did unphrased previews, which in turn led to better learning than did badly phrased previews. The factor important to determination of phrasing appeared to be the relative duration of pauses rather than their absolute duration. A second experiment was performed which confirmed that the effect of phrasing was, in the main, "to mark the natural boundaries between sub-patterns". It cannot simply be assumed that these results, which relate to the processing of visual patterns, apply equally to auditory pattern perception. Fortunately, there exist a number of studies which suggest a comparable effect in the processing of auditory patterns.

Deutsch [38], in a study which preceded Deutsch and Feroe 's paper but used as its

basis the theory of serial pattern representation developed in the latter paper, performed an experiment to relate the temporal structuring of a patterned sequence to its susceptibility to immediate recall. Four structured sequences, each twelve notes long and each consisting of four sub-sequences of three notes each, were employed together with four "haphazard" sequences derived from them, for which no pattern was apparent. Each of the eight sequences was presented in three temporal conditions: one in which all notes were equally spaced in time; one in which the tones were grouped into four groups of three; the third in which the notes were grouped into three groups of four. An analysis of the recall results showed that for sequences segmented in accordance with structure, the recall was extremely good, and slightly better than recall of the same sequences with no temporal segmentation. In contrast, the recall of sequences whose temporal grouping was at odds with their structure was poor, as was the recall of unstructured sequences. Further analysis of the "serial position curves" implied that "the temporal groups tend to be coded as units or chunks and to be retained or lost independently". A second experiment controlled against the possibility that the preference for sequences segmented as four groups of three was simply due to an overwhelming preference for groups of this size.

The results of Deutsch's experiment, namely that melodies are best recalled when their temporal segregation is "in phase" with their melodic structure, are essentially confirmed in Drake, Dowling and Palmer [51]. Although this paper is ostensibly concerned with the effect of three types of accent structure (metric, rhythmic and melodic) on the reproduction of simple melodies, its main finding was that recall was poorest when the rhythmic grouping was in conflict with the melodic grouping. As in the Deutsch experiment, rhythmic grouping was defined in terms of group boundaries which fell between successive notes of relatively long inter-onset-interval, whilst melodic grouping was assumed to be consistent with the findings of Thomassen [148] and others.

It is interesting that Deutsch often compares the results of these experiments with results of experiments using different stimulus types, including the visual stimuli of Restle's experiments (above) and the verbal/letter type stimuli used in the paper described next.

Bower and Springston [17] used letter series to test the hypothesis that "recoding" or "chunking" occurs predominantly at pauses. In a paper which has very obvious parallels with [38], they describe three experiments to test this hypothesis. They used letter series

which they believed would be naturally chunkable by one of two very similar processes: firstly they assembled 12-letter sequences which could naturally be divided into four pronounceable, but meaningless, trigrams e.g. the sequence"DATBECJAXPEL"; secondly they assembled 12-letter sequences consisting of four meaningful trigrams e.g the sequence "FBIPHDTWAIBM". In the first experiment, they presented Subjects with four types of sequence: "meaningful" sequences, "pronounceable" sequences, "non-meaningful" sequences and "non-pronounceable" sequences. The latter two types were generated from the former two types by taking the last letter of the sequence and moving it to the front. Thus, using the examples cited previously, the sequences "LDATBECJAXPE" and "MFBIPHDTWAIB" would be generated. This means of generating "non-optimal" sequences has the advantage that it leaves many of the letter-to-letter transitions unchanged, and thus one can indirectly test for the possibility that sequences are remembered in terms of these transitions. All four types of sequence were temporally grouped (using pauses) into four groups of three letters each. The results of immediate recall tests showed that "by shifting the location of the last letter, recall of optimal sequences deteriorates about 25%".

A second experiment added new control conditions, a condition with no temporal grouping and a condition in which sequences were temporally grouped non-optimally in a (4,3,3,2) pattern. The results of this experiment essentially confirmed the findings of the first experiment, placing recall levels for the new conditions between those of the original optimal and letter-shifted conditions. There followed further analysis of the data from these experiments in which graphs were produced showing the Transition Error Probabilities (TEP) and Transition Shift Probabilities (TSP) for the sequences recalled. These graphs show that Subjects were not chunking by meaning or pronounceability across pauses, though in cases like sequences beginning "IBMP" there was some tendency to code this as a combination of a meaningful trigram and a single letter. A third experiment was performed, using quadrigrams such as "YMCA", to show that the earlier results did not rely on some special quality of trigrams and that the "best" pause structure was that which corresponded to the sequence chunks. The authors concluded that "recall is best when the perceptual (pause) units correspond to the recodable linguistic segments".

The paper ends with a brief discussion concerning possible mechanisms for recoding: the authors postulate some kind of "dictionary look-up" process to establish the recodability of

a sub-sequence, which, if successful, would allow the sub-sequence to be replaced in short-term memory by a more concise code. This attractive notion is somewhat hampered by the vagueness of concepts such as "dictionary look-up" and one of the major themes of this thesis will be to introduce plausible "neural" mechanisms to accomplish such processes. The authors postulate that pauses in a sequence are the "most likely times within an input series at which the recoding search routine will be activated" and thus the distribution of pauses in a sequence will interact with the sequence structure in the ways described above. The distribution of pauses in a musical sequence and the effect that this has on the coding of that sequence will also be discussed later in this thesis.

Having detailed one of many linguistic analogues, I now return to related studies in music psychology. Dowling [50] performed a simple experiment to demonstrate "the effects of rhythmic grouping on chunking and memory for non-verbal materials, namely brief melodies". On each trial he presented Subjects with a 20-note sequence, rhythmically divided into four five-note sub-sequences, each played in the rhythm "dot dot dot dot dash". After a short pause Subjects heard a test sequence and their task was to say whether the test was contained (in exactly the same form) in the previous sequence. Approximately half of the test sequences had been contained within the list, the remainder being referred to as "noise" items, whilst approximately half of the test sequences were presented in the "dotdotdotdotdash" rhythm (called "within" tests) and the remainder in the "dot dot dash dot dot" rhythm (called "across" tests). Thus there were four conditions: across, within, across-noise, within-noise for which the correct responses would be yes, yes, no, no respectively. The results, given in terms of the probability of correct response, showed that "within" tests were significantly easier than "across" tests. This confirms the expectation that rhythmic grouping interacts with the memory of tone sequences in much the same way as with the letter sequences described above. Further analysis showed that performance on this task was better when the sequences were presented in a slow condition (3 tones/sec) than when they were presented in a fast condition (6 tones/sec) and, moreover, that in the fast condition musical training was found to be more useful than in the slow condition. Dowling postulates that both these observations are consistent with an "encoding" process which takes a finite amount of time, this being smaller for the more experienced Subjects. Finally, he also notes that Subjects showed some "adaptation" to particular rates of presentation, finding most difficult the tests which, by chance,

lay immediately after rate changes. Effects similar to this rate perseveration, and a possible mechanism to help explain it, will be studied in Chapter 8.

Tan, Aiello and Bever [146] performed a similar experiment in which Subjects were presented with brief melodies which were devised, with reference to standard music theory, so as to be most naturally perceived as consisting of two phrases separated by a melodic/harmonic boundary. In this case there was no variation in the length of the notes and therefore there was no cue to rhythmic grouping. Subjects were then presented with a two-note probe and were asked to assess whether or not the probe had been contained within the melody previously presented. Probes were of three types: the two notes which had ended the first phrase; the two notes which had begun the second phrase; and the last note of the first phrase coupled with the first note of the second phrase. The former two probe types correspond with the "within" condition and the latter probe type corresponds with the "across" condition of the Dowling experiment above. The results, which the authors point out correspond with a large body of similar experiments conducted with linguistic stimuli, show that performance was significantly worse on between-phrase probes than on within-phrase probes. This effect was found to be more pronounced for musicians than for non-musicians, a fact which the authors claim indicates that the perceived strength of the phrase boundary depended somewhat on the Subjects' (learned) ability to infer melodic/harmonic closure, an ability which was assumed to be more highly developed in musicians. This conclusion was strengthened by the observation that musicians had a pattern of response which appeared sensitive to the type of harmonic closure (cadence or semi-cadence) which marked the phrase boundary. After recapping the similarities between these results and results found in experiments on linguistic phrase perception, the authors stress their belief that "there are general processes that operate independently of modality with natural structures". Furthermore, the findings with regard to performance distinctions between musicians and non-musicians, when coupled with the letter series experiments of Bower and Springston et al. and the other studies mentioned above, suggest that the parsing of melodic sequences is not just the result of innate grouping mechanisms but very much affected by the learning that results from long-term exposure to a particular musical style. They suggest that in melody perception, as in linguistic perception, the "chunking" of typical or familiar sub-sequences, influenced as it is by both past experience and by current temporal grouping, plays an important part in the subjective structuring and

the memory of extended sequences.

This notion bears a striking resemblance to the view proposed by Oura [119]. In this paper the author postulates a "reduced-pitch-pattern model for melodic processing" in which "prototypes", consisting of archetypal melodic patterns, are stored in long-term memory and are used to aid in the memorizing of new melodies. These prototypes are composed of what Oura calls the "structural notes" of melodies, and are distilled from previous exposure to melodies of similar style. The process of memorizing melodies in a particular style is then reduced to the parsing of the melody into segments out of which the structural notes are abstracted. These resulting pitches, now devoid of their rhythmic information, are "matched", if possible, with the learned prototypes in long-term memory, leading to enhanced memorability. In the case of no exact match being found, the current set of structural notes can either be stored in long-term memory as "an exception" (and possibly, following repeated exposure, a new prototype) or, Oura suggests, as a combination of a prototype acted on by relevant additional modifiers. The melodic ornamentation and rhythmic information present in the real melody is also presumed to be stored in modifiers separate from the tonal prototype. The use of such modifiers is to prevent proliferation of prototypes describing, for instance, the same group of structural tones in a multitude of different rhythmic configurations (see Chapter 8 for a discussion of the "factoring out" of rhythmic information). Oura further proposes that the enhanced memory for melodies of a particular style, demonstrated by Subjects most familiar with that style, can be explained by the relative abundance of prototypes and modifiers available to those with most relevant experience.

The viability of this model is then compared with that of two alternative models, one based on contour, the other on underlying harmony, with reference to an experiment performed to test the memory of experienced musicians for a melody which, though unfamiliar, was in a familiar style. The results indicate that musically experienced listeners do not, as noted previously in this chapter, rely on contour to memorize extended melodies. Furthermore, an analysis of the errors made in performing the memory task indicate that the proposed reduced-pitch-pattern model is superior to the model based on consideration of the underlying harmony alone, though it does not exclude the latter as a contributory mechanism. The basic idea presented in this paper i.e. that a melody in a given style can be stored as a combination of prototypical patterns, stored in long-term memory as a result of prior exposure to pieces

in that style, is an idea which gives strong and independent support to many of the concepts suggested in this thesis. Sloboda and Parker [144] draw similar conclusions from an experiment which tested the immediate recall of melodies. They summarize: "Musicians code harmonic relationships that seem less accessible to non-musicians. In both groups however there is evidence of a great amount of common processing. Subjects seem to share the pool of basic melodic and rhythmic building blocks ...".

2.3.4 Summary

The parsing of musical sequences, melodic and harmonic, can be viewed in the light of a more general ability, which also manifests itself in visual and linguistic tasks. In the case of visual pattern learning, theories have been developed relating the ease of learning a given pattern to the ability of that pattern to be generated from a hierarchical rule system. Such an explanatory mechanism is not appropriate for verbal/letter series tasks, for which concepts of pattern familiarity and meaning are therefore introduced. Serial patterns in music have also been represented as hierarchical encodings but these, though mathematically neat, do not seem to relate sufficiently to the realities of *perception* and the rôle of prior learning.

A different perspective, the one which will be adopted in this thesis, is that the division of melodic/harmonic sequences into phrases, is mediated not only by primitive perceptual grouping mechanisms but also by the influence of prototypical sequences, stored in long-term memory, learned by exposure. The prior acquisition of such prototypes confers processing and memory advantages on an individual in response to sequential stimuli of the relevant type. If these prototypes are represented in LTM as typical sequences of scale-notes/chord functions, they will embody much of the information implicit in the idea of a tonal framework.

The next section will review aspects of musical expectation.

2.4 Expectation in Music Psychology

2.4.1 Review

As detailed in the previous chapter, the concept of expectation is widespread in music theory. In this section I will review some of the theoretical and experimental contributions made to

the study of musical expectation by psychologists. Once again I shall highlight those points which are particularly relevant to issues raised later in this thesis.

Carlsen [20] investigated the influence of various factors on the melodic expectancies elicited from a group of Subjects. Working from the (usual) assumption "that expectancies are a function of a person's previous experience with events", he devised an experiment to test the effect of three factors on Subjects' melodic expectations. Firstly, he selected Subjects from three different countries (Germany, Hungary and the U.S.A.) in an attempt to ascertain the effect, if any, of cultural milieu. This choice of countries was influenced by earlier observations which had indicated that the song repertoire of Hungarian schools contained melodic patterns which differed significantly from those prevalent in the songs most often performed in American schools. A priori, it seemed possible that the German Subjects might exhibit responses which were in some sense intermediate. The second factor considered was the Subject's degree of musical training—it was hypothesized that Subjects trained to a professional level would have been exposed to a wide repertoire of music, perhaps sufficient to override, to some extent, the influence of their earlier musical experience. To investigate this hypothesis, Subjects formed two groups: one group drawn from music students in the earliest stages of their training; the other consisting of persons enrolled in advanced professional music schools. Lastly, the influence of a Subject's vocal register was considered, the hypothesis being that the melodic patterns typically encountered by a singer in a particular register, e.g. bass, would be significantly different from those encountered by a singer in another register, e.g. soprano, and that this difference might well affect a Subject's melodic expectations accordingly.

During the experiment Subjects were presented with a "melodic beginning" drawn from the set of 25 two-pitch intervals found within the equally tempered chromatic scale (12 ascending intervals, 12 descending, 1 unison). The Subjects' task was to sing the expected continuation of the melody, in the rhythm expected, using the tempo indicated by an accompanying metronome. Enough time was given for Subjects to sing a short musical phrase. In this way, 91 Subjects produced a total of over 34,000 sung responses, which were subsequently analysed with regard to the first sung pitch only, to produce a sample of response intervals (being the interval between the last stimulus tone and the first response note). Preliminary inspection of the results showed not only that, as expected, different melodic beginnings elicited different expectations, but also that different melodic beginnings elicited expectancies of varying

strength, measured in terms of the proportion of times a particular response interval succeeded a given stimulus interval. Perhaps surprisingly, both level of musical training and vocal register had no significant effect on the expectancy patterns generated. In contrast, the results indicate a considerable influence of cultural milieu—16 of the 25 possible beginnings elicited significant expectancy differences, which could be attributed to a Subject's cultural background. Further analysis showed that some of the three-note patterns produced could be directly related to harmonic triads in various inversions. Major and minor triads were evident but not diminished or augmented triads. In addition, in spite of the instructions to continue the melodic beginnings by singing a series of notes (as opposed to just one note), a directive which it was hoped would reduce the otherwise high proportion of patterns interpretable as full or half cadences, certain of the three-note patterns showed clear cadential influences. Carlsen also highlights patterns in which the response interval is the inverse of the stimulus interval (cf. Narmour's "registral return") and in which the response interval is unison, but points out that none of the above "explanations" can shed more than limited light on the pattern of responses achieved.

In summary, Carlsen's experiment affords important experimental data on the factors influencing expectancies and concludes that, from among the possible factors studied, these expectancies are heavily influenced by a Subject's cultural background, but not significantly by a Subject's vocal range or level of professional music training.

Schmuckler [140] begins his paper with a review which emphasizes the importance of the study of expectation in other areas of psychology ("perception, attention, learning, language processing and so on"), as well as in music psychology. He cites Carlsen [20] but suggests a number of problems inherent in Carlsen's procedure: firstly, that Carlsen's experiment was only able to test for conscious or explicit expectations but not for implicit expectations ([109]); secondly, that the two-note stimuli may not have been sufficient to generate strong expectations. He therefore conducted four experiments, in which he investigated the melodic and harmonic expectancies generated by, respectively, the melody and piano accompaniment of a Schumann song presented both separately and together as part of a full musical context.

The procedure was similar for the first three experiments. Stimuli were derived from the first eight measures of the Schumann song, in which had been selected 10 stopping points ("probe positions"). Subjects were presented with ten blocks of trials, each block consisting of

stimuli which stopped at a particular probe position, taken in the order of their occurrence in the piece. For each block, and therefore for each probe position, Subjects were presented with the melodic/harmonic fragment which began at the start of the piece and ended immediately before the probe position. At this point they indicated the strength and specificity of the resultant expectation by rating each on a 7-point scale. Then the Subjects were presented with a set of experimental trials in each of which the fragment was augmented by a single event. For each trial Subjects were to assess, again on a 7-point scale, the continuation event in terms of how well it matched their expectation. It was made clear that this assessment should "reflect how well the continuation fit in with the ongoing process, as opposed to how well it functioned as a final 'resting' position".

In the first experiment, melodic fragments were presented, together with a set of continuation tones consisting of the one octave chromatic scale centred around the true continuation note. Each continuation tone had the same duration as the true continuation note. Schmuckler compared the results of this experiment with those expectations indicated by three different music-theoretic views of music structure. The first of these was concerned with the influence of tonality, i.e. the perceived stability of certain scale degrees (particularly the tonic, sub-dominant and dominant) within a tonal context. The second was concerned with the melodic contour of the passage: it was hypothesized that Subjects might expect notes which preserved the contour direction. Thirdly, it was suggested that the concept of a "melodic process" ([110]) might be instrumental in the establishment of expectancies. These melodic processes are archetypal melodic motions apparent in the melodies of a given culture and presumably familiar to those immersed in that culture. An analysis of the results showed "a significantly positive correlation [indicating] that the highly expected tones are the same as the psychologically stable tones within a key" and thus strong support for the influence of tonal structure on expectancy formation. Regarding the melodic contour, the results indicated a significant preference for contour-changing continuations in contradiction of the original hypothesis, though little effect of contour was evident overall. The influence of melodic process was, however, found to be great; in each probe position the expectancy rating for the tone suggested by such an analysis was significantly greater than the average rating for the remaining diatonic tones. A regression analysis confirmed that tonality and melodic process dominated expectancy formation whilst contour never made a significant contribution.

Context chord	Often Follows	Sometimes Follows	Seldom Follows	Not Considered
I	IV,V	VI	II,III	VII
II	V	VI	I,III,IV	VII
III	VI	IV	II,V	VII,I
IV	V	I,II	III,IV	VII
V	I	IV,VI	II,III	VII
VI	II,V	III,IV	I	VII
VII	III	—	—	I,II,IV,V,VI

Table 2.1: Usual Root Progressions (after Piston (1978))

The second experiment was similar in structure to the first. In this case the stimuli consisted of the relevant portion of the song's harmonic accompaniment and the set of continuations consisted of the "seven chords built on the members of the diatonic scale", the exact diatonic scale being specified by the presumed key (tonal centre) of the piece immediately before the probe position. As before, the continuations preserved the rhythmic qualities of the true continuation chord. The expectancy ratings for chords were compared with theoretical ratings derived from a "Table of Usual Root Progressions" ([123]) (see Table 2.1).

These theoretical ratings of expectancy tacitly assume that the particular archetypal chord sequences suggested by Piston's analysis, which itself is derived from analysis of typical progressions in the western tonal/harmonic style, have been internalized by the Subjects during exposure to such a musical environment and are available to them for expectancy formation. The results indicate that this assumption is justified in general, in that "common chord progressions were highly expected". The correlation between predictions and results was not complete, however, and Schmuckler seeks to explain this by pointing out that Piston's table only provides for a limited context (one chord) and furthermore that the table might decrease in its reliability for the lower levels of expectation.

The third experiment employed as stimulus the full musical context i.e. the melody together with its accompaniment. The set of continuations consisted of 7 possible melodic continuations, being each of the notes in the diatonic scale, each harmonized in three different ways, as the tonic, third or fifth of the triad. This gives 21 possible continuations for each

probe position. By analysing the results of this experiment together with those of the first two experiments, Schmuckler was able to demonstrate apparent perceptual independence between melody and harmony in their generation of expectancies. In other words, melodic and harmonic factors combine additively in expectancy formation. It is worth noting here that in all of the first three experiments the ratings for expectancy strength and specificity are highly correlated and that the expectancy rating for the true continuation tone/chord was significantly greater than the average expectancy rating in all probe positions except one. It was pointed out that the exceptional probe position marked what was (a priori) a "highly improbable" event in the piece.

The fourth experiment was reminiscent of Carlsen's experiment and involved a production task. Pianists were asked to produce melodic and full (melodic/harmonic) endings to the musical fragments presented in the first three experiments. They were allowed to rehearse their productions, which could be of any reasonable length. The data not only indicated that the first notes/chords of their productions were highly correlated with the previously derived expectancy profiles but that the average length of their productions was negatively correlated with the strength/specificity ratings of those expectancies i.e. events affording strong/specific expectations elicited shorter completions.

Schmuckler [141] provides further analysis of the data obtained in experiment 4, which suggests that expectancies, as indicated by the Subjects' improvisations, appear to be "referent-guided" i.e. guided by underlying schemata. He notes the distinction between "in-time" referents (musical themes, motives etc.) and "out-of-time" referents (moods, emotions etc.) but points out the difficulty of distinguishing experimentally the effect of each. Schmuckler also concludes that expectancies can be formed at the phrase level, each phrase comprising a music "chunk".

To summarize Schmuckler's important papers, Subjects were shown to generate expectancies of varying but correlated strength and specificity in response to melodic, harmonic and melodic/harmonic stimuli. In the case of melody, these expectancies were seen to be consistent with explanations based on familiarity with a tonal structure and with archetypal melodic processes. In the case of harmony, expectancies were consistent with music-theoretic data relating to the environmental (cultural) frequency of certain chord combinations. There appeared to be perceptual independence between these two types of expectancy. Taking these conclusions

together with those of the Carlsen paper, a picture emerges in which Subjects' expectancies, and by extension the events they "produce" when asked, are very strongly influenced by the internalized regularities of their musical environment.

Kalmar and Balasko [90] reinforce this conclusion in a study of the improvisations of Hungarian school children, in which they conclude that their results "support the assumption concerning the existence of a 'musical mother tongue'". By this they mean that even young children had internalized the typical features of the Hungarian folk/children's songs to which they had been repeatedly exposed, and that they were able to use this knowledge to generate expectancies and thus plausible improvisations.

In a similar vein, Abe and Hoshino [1] performed an experiment to investigate the final-tone extrapolation behaviour of two Subjects, one an expert in western classical music, the other an expert in traditional Japanese music. The results indicated that the "experts' music cognition is highly schema-driven", but that each Subject employed different schemata, related to their musical background. In particular, the responses of the western expert were strongly influenced by the western diatonic tonal framework, although the authors point out that the *dynamic* aspects of this influence remain to be investigated.

Another set of experiments viz. Dowling [43], Dowling, Lung and Herrbold [49] and Andrews and Dowling [4], approach the idea of expectancy in a different way. The first of these papers refers to a set of experiments performed with "interleaved melodies", in which Dowling discovered that, for Subjects to identify two familiar melodies whose notes had been interleaved (in a regular alternate manner), it was necessary that the melodies were transposed so that the pitch ranges of each melody were separated by, on average, just over a semitone. A second experiment demonstrated that the interleaving of an unfamiliar background melody interfered somewhat with the short-term recognition of another previously unfamiliar melody but that this interference was again minimized by pitch separation. The next experiment demonstrated that Subjects were able accurately to confirm the presence of a previously named familiar melody in conditions in which it was interleaved with a background melody of overlapping pitch range. In "catch trials" where the melody was either an unfamiliar melody or a familiar melody different from the target, Subjects were unable to recognize any pattern in the stimulus. Thus it appears that Subjects were able to "search" the stimuli for expected notes in order to confirm or deny their presence. Dowling extends this idea to conclude that

"the expectations of the listener who knows the structural principles of a form will lead him to hear different things in a piece than will be heard by the naive listener".

The second study, Dowling, Lung and Herrbold [49], introduces the concept of "expectancy windows", areas in pitch-space and time within which events could be perceived accurately at the expense of other excluded events. The authors relate this concept to other evidence demonstrating that expected events are processed more efficiently than unexpected events in a variety of modalities. The results of various experiments are given which serve to clarify the concept of "expectancy windows". The first of these was partially a repetition of Dowling's [43] third experiment (above). It confirmed that when distractor tones were separated in pitch range from the target melody, the presence/absence of the named melody was easy to assess. However, it was further discovered that when the two melodies were in overlapping pitch ranges, performance was better when the notes of the target fell on the (implicit) beat than when they fell off-beat. This result suggested that expectancy windows could be most easily "anchored" at prominent points in the temporal sequence.

A second experiment employed a "moving tone" which, though it occurred at an expected time, had a variable pitch. Subjects' perception of this pitch was assessed using both a pitch-based and contour-based judgement task and it was shown that performance in these tasks was significantly better when the moving note fell within its expected pitch range than when it was remote from this region. This confirmed the hypothesis that the expectancy windows were limited in pitch-space. Experiments 3 and 4 essentially repeated experiments 1 and 2 but did not include distractor tones. This had a dramatic effect—the removal of distractor tones appeared to remove the need to focus attention on expectancy windows at all and remote notes were processed just as efficiently as notes within the expected region. Finally, further experiments suggested that Subjects were able to focus on expectancy windows which were centred off the beat (i.e. on expected syncopations), though not with the same level of performance as when the expectancies were directed towards the more prominent beat.

The third paper in the series, Andrews and Dowling [4], further extends the concepts introduced in the first two, with particular emphasis on the development with age of control of auditory attention. It also proposes useful definitions of the terms "attention" and "expectancy". Attention is considered to refer to "processes by which a listener selects a stimulus 'stream' or 'channel' on the basis of stimulus features such as pitch, loudness, timbre and di-

rection to source". These processes are essentially the same as those primitive processes which contribute to Bregman's "auditory stream segregation" [18]. Stimuli in a given stream are able to be processed and the pattern that they form can automatically activate appropriate representations stored in long-term memory (if they exist, e.g. patterns representing familiar melodies) without priming these representations from the top-down by, for instance, prior naming. By contrast, expectancies operate "when the listener is cued concerning a small set of possible targets and has memory representations of those targets. Expectancy is the process by which the listener succeeds in selecting potential target elements and evaluating them against memory representations". The authors propose that, in the absence of the attentional streaming described above (as in the interleaved melody task with overlapping pitch ranges), listeners are guided by expectancies which allow them to confirm or deny the presence of a primed familiar target pattern. The expectancies are focused on specific times and specific areas of pitch space and events falling outside of these areas are not processed at all.

The experiment performed to evaluate the usefulness of these definitions is a complex one, measuring the importance of each of these processes for Subjects of various ages in an attempt to chart the development of various abilities. The results, and the conclusions drawn from them, are numerous but it is worth noting one in particular: Andrews and Dowling suggest that even experienced adults found it difficult to focus attention rhythmically in time, in the sense that they appeared unable, at the presentation rates employed (6 notes/second), to "pick up whatever stimuli occurred in the critical time slots". In contrast, both adults and children were able to focus expectancies in time, i.e. they were able to ascertain whether or not certain expected events occurred at those critical time points.

This discussion of the spatial/rhythmic nature of attention and expectancy dovetails well with the influential work of Jones, aspects of which will be discussed next.

Jones [84][85], in a two-part essay, draws on concepts introduced in [83] in order to develop a model of musical expectancy. This model revolves around her assertion that expectancies must be based on temporal as well as spatial structure i.e. expectancies contain elements of both "what" is expected and "when" it is expected to occur. She claims that regularities in a stimulus elicit a vector of expectancies with melodic, harmonic and rhythmic dimensions. "Experienced" listeners are attuned to these regularities by virtue of their increased previous exposure to music in the relevant style. To quote Jones, "...an expectancy captures that

private sense of anticipation experienced with an unfolding pattern ...".

In further developing these ideas she introduces the notion of Ideal Prototype patterns and Ordinary patterns. Ideal Prototype patterns are the generic or invariant structures of a particular style whereas Ordinary patterns are the real musical stimuli which are "more or less artful variations" of these archetypes. In the western idiom, for example, Ideal Tonal/Melodic patterns correspond to well-structured melodies derived from the tonal scale alphabet (as in [41], etc.) whilst Ideal Time patterns correspond to the rhythm and tempo of conventional music. These Ideals provide a context against which Ordinary Patterns can be developed i.e. the irregularities or "surprises" which occur in real musical pieces are evaluated by their relationship to "perceptual reference frames" established by expectancy schemata derived from Ideal Prototypes. Skilled listeners are assumed to have more experience with a given genre and thus to have a larger array of Ideal Prototypes at their disposal for use in interpreting, for example, novel melodies. Jones completes her analysis with a short review of the rôle of harmonic, melodic and rhythmic context, and by extension the corresponding expectancies elicited by this context, on the perception of musical stimuli, as manifested in recent experiments. In general, Subjects' performance at various types of musical judgement task was shown to improve at points where the context most strongly activated expectancy schemata, which in turn provided the listener with well-defined perceptual frames.

Jones' subsequent work is concerned not just with expectation but with a dynamic model of music perception which seeks to incorporate aspects of melody, harmony, rhythm and metre into a more general, time-based, theory of attending. I will return to this work in Chapter 8.

I shall end this section on musical expectation by referring to a series of papers written by Bharucha et al.. These papers develop a model of expectation that is based on a "connectionist" framework in which expectancies are seen as being equivalent to the "priming" of neuron-like units by the spreading of activation through a network of such units.

Bharucha and Stoeckig [11] introduce "a spreading activation model of the representation and processing of harmony". Their model is based on a layered network of nodes: one layer is made up of nodes representing Major chords; another layer is made up of nodes representing Minor chords; a third layer consists of nodes representing the Major keys. Links are made between layers such that chords are linked to their "parent" keys. The strength of these links represents the "degree to which the chord establishes the parent key". It is important to

note that in the networks discussed by Bharucha and Stoeckig these link strengths are set by hand, i.e. the networks are hard-wired by an "expert"; this contrasts with their assertion that an equivalent network can "self-organize" in response to the relevant musical environment. When a chord is sounded the corresponding node is activated and this activation is allowed to spread across the network, activation spreading most easily along strong links. The overall activity level is assumed to decay over time. The authors postulate that harmonic expectations result from chord units being indirectly activated, i.e. activated by virtue of the spreading activation rather than by virtue of the corresponding chord having been sounded (N.B. in the model shown there does not appear to be any way of distinguishing between these two states). Furthermore, they hypothesize that such prior activation can facilitate the processing of an expected chord and that this facilitation would manifest itself as an improvement in the time of reaction to the expected stimulus.

They performed three reaction-time experiments in which musically trained Subjects were presented with a chord (the prime) as context followed by a second chord (the target). The prime and the target were either harmonically related, in the sense that the model developed above would predict that the prime would elicit expectancy for the target, or they were harmonically unrelated. Subjects were asked to perform a true/false decision task in response to the prime-target pair, and it was hypothesized that faster decisions would be made or related targets. In the first experiment the task was to make a major/minor decision for the target chord. The results indicate that major targets were identified much faster and much more accurately when they were related than unrelated. Interestingly, however, the reverse was true for minor targets, though the difference in this case was not significant. Bharucha and Stoeckig explain these results by pointing out that they are consistent with a systematic bias for judging related targets to be major and unrelated targets to be minor. They account for this bias by citing evidence that major chords are perceived to be intrinsically more stable or consonant than minor chords. In the experiment the prime is assumed to increase the perceived consonance of the related targets, thus accounting for Subjects' tendency to label such targets as major chords. In other words, major chords are, by nature, more consonant therefore more consonant chords are more likely to be labelled as major.

This hypothesis is investigated in two further experiments in which the major/minor decision is replaced by a well-tuned/ill-tuned decision. Mistuned target chords ("foils") were

generated by shifting the pitch of one of component notes of the corresponding well-tuned targets by an eighth of a tone. Major targets were used in one experiment and minor targets in the other and in each case the foils were less consonant than the well-tuned targets, thus avoiding the problems described above. The results indicate the priming of minor chords and thus support the explanation given for the lack of such priming in the first experiment. It is worth noting that the in-tune/out-of-tune decision task used in the second experiment is within the capability of musically untrained Subjects (unlike the major/minor decision task) and therefore for this experiment both trained and untrained Subjects were employed. The results show that there was no significant correlation between the strength of priming and the level of musical training, suggesting that the expectancies which lead to a priming effect are learnt by exposure rather than by specific training.

Bharucha and Stoeckig [12] performed two follow-up experiments which eliminated a possible explanation for these results based on the hypothesis that shared tones between the prime and the target were responsible for the priming effect. In these experiments they showed that the priming effect was evident when the prime and target had no tones in common and even persisted when all the overlapping components of the harmonic spectra of the two chords were removed.

These results, the authors suggest, indicate that the priming effect is the result of a cognitive process, rather than resulting from a simple sensory mechanism. They go on to state that "...In the absence of evidence that this internalized structure is innate, we assume that it is the consequence of a general-purpose connectionist architecture that internalizes regularities in the environment ... the internalization of musical structure does not come from explicit musical training, but presumably from mere exposure". This thesis proposes the use of a just such a general-purpose connectionist framework to model these and other results, though the framework developed here differs in important ways from that suggested by Bharucha et al.. Most importantly, I shall show how an initially "ignorant" network can organize itself in response to environmental stimuli in such a way that it can internalize regularities within that environment in the manner described above.

Bharucha [9] reviews the Bharucha and Stoeckig papers and describes a series of experiments, carried out with Indian Subjects, designed to test the priming effect as it manifests itself in a musical tradition which is essentially "non-harmonic". He hypothesized that "a

tone that is considered important in a particular rag would be facilitated even if it doesn't appear in the experimental context". In order to test this hypothesis, Subjects were played a traditional rag, in which one of the typical melodic figures was left incomplete (unresolved). Immediately afterwards they were asked to perform a task involving an in-tune/out-of-tune decision similar to the one above. The results show that the tone which would typically complete the figure was primed even under rigorous conditions in which overlapping portions of relevant harmonic spectra were removed. Bharucha concludes from these experiments that "...a musical context activates mental representations of musical regularities. Tones that are absent from the context but typically follow are facilitated by top-down activations from these representations ...". It is not clear how a network such as the one developed by Bharucha, in which tone units are connected to higher-order units which respond to simultaneous (or near simultaneous) sounding of those tones i.e. chord units, could account for the priming effect elicited by the unresolved rags. In contrast, I shall develop a model in which connections are established between tone units (or chord units) and higher-order units which represent typical *sequences* of tones (or chords). In this model expectancies will rely on the activation of nodes representing archetypal sequences preceding the full statement of the corresponding stimulus pattern. The difference between the two models is highlighted by comments made by Bharucha [10] with reference to the ability of his network (which he calls **MUSACT**) to self-organize in response to a musical environment using a "competitive learning" (CL) algorithm. In his words, "the repeated occurrence of major and minor triads ... would result in the specialization of units that respond to particular major and minor chords ...". With this I am in agreement (although I would hesitate to use the CL algorithm for reasons given in the next chapter). However, he goes on to say "...Similarly, the regular occurrence of chords in typical combinations would result in the specialization of units that respond to particular keys ...". This appears to be a non-sequitur. I would suggest that, in certain circumstances, the regular occurrence of chords in typical combinations would result in the specialization of units that respond to *those combinations*. In short, I believe that Bharucha's error lies in his belief that competitive learning can lead to a connection pattern in which *many different patterns* of activation (each representing, for instance, a chord progression characteristic of a particular key) can activate a single higher-level node (corresponding to the relevant key), without that node also being activated by other spurious and possibly atypical patterns. This

observation has, I believe, serious consequences regarding the ability of Bharucha's network to self-organize in the way described. These problems are not shared by my model, which specifically attempts to form higher-level nodes that respond to typical sequences. Additionally, the model I shall develop will cater for the recognition (bottom-up processing) and production (top-down processing) of familiar sequences e.g named melodies, and the so-called "veridical" expectancies which these elicit—Bharucha lists the lack of this feature as one of the limitations of his model.

Bharucha [10] does go on to describe a different network for the learning of sequences, based on networks found in Jordan [88]. Discussion of this network will be found in the next chapter.

Finally, returning to the experiments of Bharucha and Stoeckig, I shall present an alternative analysis which emphasizes the assumed *function* of the priming chord and the rôle which this function has in generating expectancies. By "function" I mean the Roman numeral description of chords, which relates each chord to the tonic of the current key. Taking both sets of experiments as a whole, we see that for a C Major prime the chords designated as "related" are F Major, G Major, A minor and Bflat Major. If we make the assumption that the C Major chord suggests the key of C Major, then the first three prime-target progressions can be thought of as I-IV, I-V, I-VI progressions. Referring to Piston's Table of Usual Root Progressions (see Table 2.1) we see that these progressions are highly/moderately likely to occur in western harmonic music. The fourth progression (C Major-Bflat Major) cannot be written in functional form if we consider the key to be C Major. However, if we assume that C Major is acting as the dominant triad in the key of F Major, this progression can be viewed as the highly likely V-IV progression. Data from Krumhansl and Kessler (1982) indicate that these key assumptions are indeed those most likely to be made by Subjects in response to the single chord C Major. By contrast, none of the "unrelated" progressions can be written in functional form.

A similar analysis can be carried out on the experiments which employed an A minor prime, the only exception being the A minor-C major progression which, though it can be written as a number of possible functional progressions (e.g. VI-I in C Major), cannot be written as a functional progression which could be considered likely by reference to Piston's Table. An obvious explanation for the fact that A minor primes C Major is that, as a relative

minor/major pair, the two chords have two notes in common.

Of course this functional analysis begs the question of how the plausible keys are established in the first place. Krumhansl and Kessler [98] suggest that the key-assignment process is such that at any one time a transfer to a closely related key can be accomplished easily to account for stimulus events. This suggests a mechanism in which a variety of different but related candidates are being considered in parallel as the current key. The favoured key interpretation will be the one in which expectations are best confirmed by stimulus events. The idea that the current favoured key interpretation is that in which the recent events "make sense" with regard to expectations for them, is mirrored somewhat by the fact that the method used by Krumhansl and Kessler to obtain a key profile at a point in musical pattern is almost identical to the method used by Schmuckler [140] to establish an expectancy profile in similar circumstances.

Obviously the fact that Bharucha wired his network to reflect the strong links between a given key and its I, IV and V chords, means that this functional-progression analysis is hardly differentiable from his "key membership" analysis. However, in the light of my doubts concerning the ability of Bharucha's network to self-organize in the way described and my subsequent concentration on typical chord progressions, it seemed reasonable to introduce the notion of chord function at this stage. An added advantage of viewing chords (and tones) as functional relative to a tonic centre is that one can dispense, unlike Bharucha, with the need to proliferate nodes to account for all possible points along a frequency continuum.

In spite of the limitations described above, Bharucha's papers, and the models proposed therein, are both important and influential in the sense that they establish a connectionist framework as one which recommends itself to the study of musical phenomena. In the next chapter I shall review various musical applications of connectionist models.

2.4.2 Summary

Musical expectations are learnt by exposure to a musical environment—they result from a process that internalizes the regularities of that environment by representing them as a set of mental schemata. Expectations are elicited when sufficient evidence is available in a stimulus to activate one or more of these learned schemata and they can facilitate the processing of subsequent (expected) events. Listeners familiar with western tonal music form melodic,

harmonic and rhythmic expectations—there is some evidence that these expectations are perceptually independent. The idea of tonality is implicit in the expectations experienced by such listeners and this is best embodied in a functional representation of musical sequences.

Chapter 3

Neural Networks and Music

3.1 Introduction

This chapter reviews some of the ways in which neural networks have been applied to the field of music. The range of applications is large, so most emphasis will be placed on those which employ neural networks in the modelling of aspects of music perception. Much of the work discussed in this chapter was originally published in two special issues of the Computer Music Journal (Volume 13, No.3 and No.4 (1989)) and was subsequently collected in Todd and Loy [150].

A connectionist approach recommends itself to the modelling of musical "intelligence" in various ways. Firstly, connectionist models acquire knowledge simply by exposure to a given "environment", and as such reflect the real-life process of, in particular, *musical* knowledge acquisition more closely than do traditional AI techniques, such as those employing "expert" rule systems or statistical methods. Secondly, not only do connectionist approaches make claims to model the *process* of musical learning, but they also claim, to varying degrees, to do so by employing "hardware" bearing some resemblance to that found in the human brain. In many cases this resemblance is slight, but all connectionist models at least describe the behaviour of networks of simple units (corresponding to cells or neurons), connected together by "weighted" connections (corresponding to axons and synapses), whose weights are often able to be modified via a given "learning rule". For this reason the use of such networks is often referred to as *parallel distributed processing*. Whilst little is known about the precise way in which the human brain functions, the "biological plausibility" of connectionist models

can be assessed with reference to other yardsticks: in particular, the models developed in this thesis place considerable emphasis on the ability of the networks to "self-organize" in response to "arbitrary" real-time input environments. Many of the models discussed later in this chapter fail to account for such self-organization and employ external "supervisors", whose plausibility is highly questionable. Thirdly, music perception requires the processing of highly patterned input data, patterned in spatial-type "vertical" dimensions, such as the harmonic dimension, as well as in the temporal dimension. Neural networks perform well at such "pattern-recognition" tasks, exhibiting some desirable properties, such as the ability to generalize in response to novel input and the associated ability to degrade their responses "gracefully" when faced with fuzzy or corrupted input.

The field of music perception, in turn, offers considerable scope for the testing of connectionist models of human intelligence. Music pervades the cultures of the world and exhibits interesting variations between those cultures. There is a considerable amount of theory relating to structures inherent in the music itself and a corresponding amount of experimental and theoretical work postulating ways in which humans respond to those structures (some of which is detailed in my previous chapter). Music, in particular western tonal music, exhibits regularities which make its study more tractable than that of other apparently universal human behaviours, such as language learning, yet it is of sufficient complexity to make such study far from trivial. Indeed, it seems highly likely that certain mechanisms underlie the perception of both music and language — the models developed in later chapters certainly support this assertion — and as such the study of one may well give important insights into the study of the other.

In summary, neural networks have a great deal to offer to the modelling of music perception, whilst the field of music perception itself offers an excellent test-bed on which to develop connectionist models. In the next section I shall describe in more detail specific tasks to which such models have been applied.

3.2 Neural network applications in music

3.2.1 General review

As noted above, neural networks have been applied to a diverse range of musical problems. Todd and Loy [150] include the work of Sayegh [137], who uses an optimization network to generate "good" finger patterns for the playing of stringed instruments, and the work of Lewis [106], who employs a standard supervised network to learn to become a "critic" of its musical input, before using it in a process of "creation by refinement". This latter process is similar to the "evolutionary" system developed by Gervais and Greenhough [61], which produces musical output guided by the critical opinions of, in this case, a human listener.

Regarding models of music perception, Sano and Jenkins [136] describe a neural model that seeks to account for the mapping of the patterns of cochlear activity, that occur in response to the sounding of a single complex tone, into a representation of the tone that includes "both pitch and octave". Their network is hard-wired at nearly all points, as indeed may be the equivalent system *in vivo*, and does little more than "collapse" the high-dimensional input by passing it through a number of rather *ad hoc* processing layers. The penultimate layer rather implausibly funnels the input into cells which respond to precisely determined musical notes (e.g. $C_6\sharp$), a construction which seems to place too much emphasis on the determination of the *absolute* frequency of the input tone as it relates to a specific pitch system (the western chromatic scale at concert pitch). The provision of an "autoassociative, fully connected back propagation neural network", to clean up the resulting pattern of activity, casts further doubt on the plausibility of the model, since such supervised learning requires the availability of an "omniscient" teaching signal, which already conveys the identity of the input tone—if such "knowledge" is already locally available there seems little point in regenerating it.

Laden and Keefe [100] also use supervised back-propagation (BP) networks in the modelling of chord classification. Their initial experiments with small numbers of "hidden units" encourage them to abandon a simple pitch-class representation of the chord's constituent notes in favour of representations based on the psychoacoustic theories of Goldstein [66] and Terhardt [147]. The training of networks with patterns using the latter representations was shown to increase identification performance, not only for the training set but also for patterns representing incomplete or "inverted" inputs. Once again the ability of the networks to

self-organize is open to question, since a local omniscient supervisor is assumed.

Bharucha [14], in adopting an approach more consistent with the available psychological evidence, makes similar criticisms regarding the inappropriateness of supervised learning procedures. He postulates that supervised learning should be restricted to "tasks for which target vectors are both available and necessary", giving as an example the learning of musical sequences. In fact, the work presented in this thesis shows that the learning of sequences does not require supervised learning — the structure, and plausibility, of Bharucha and Todd's supervised network for sequence learning is discussed later in the chapter. Bharucha goes on to consider evidence relating to the appropriateness of various representations for pitch, with particular reference to the processing of pitch sequences. Briefly, he rules out a "spectral" representation alone as being "not a representation of the pitch at all but rather an elaborated representation of the signal, from which the pitch is extracted". He notes that a "pitch-height" representation not only ignores octave equivalence but is also unrealistic in its description of *absolute* pitch. An interval coding, in which the interval between successive pitches is used as input rather than the pitches themselves, exhibits transposition invariance, but contradicts the perceptual "reality" of each individual note, as well as failing to explain "tonal confusability" effects [44] and the lack of cumulative errors in sequence productions [149]. A "pitch-class" representation, in which pitch-height is mapped into a single octave, suffers from the same problems regarding absolute pitch as the representation from which it is derived. Finally, Bharucha considers an "invariant pitch-class representation" as being necessary for the long-term encoding of pitch sequences. This scheme is the same as that suggested in the previous chapter, in which pitches are represented relative to a tonal centre determined, in the western tonal idiom, by the current key.

Bharucha goes on to consider how such a pitch representation might be derived from musical input so as to generate the input to a sequential memory. An overview of his model is shown in Figure 3.1. It can be seen that a clear separation exists between the sequential memory (h) and the part of the network that establishes the key (a–e). Bharucha then concentrates on "how the mapping from c to d is learned", resulting in the connectivity assumed in his MUSACT model discussed in my previous chapter. The mechanism employs a competitive learning procedure to allow units at level d to become specialized in responding to a particular type of chordal input (e.g. major triad), but requires an unusual "yoking" of paral-

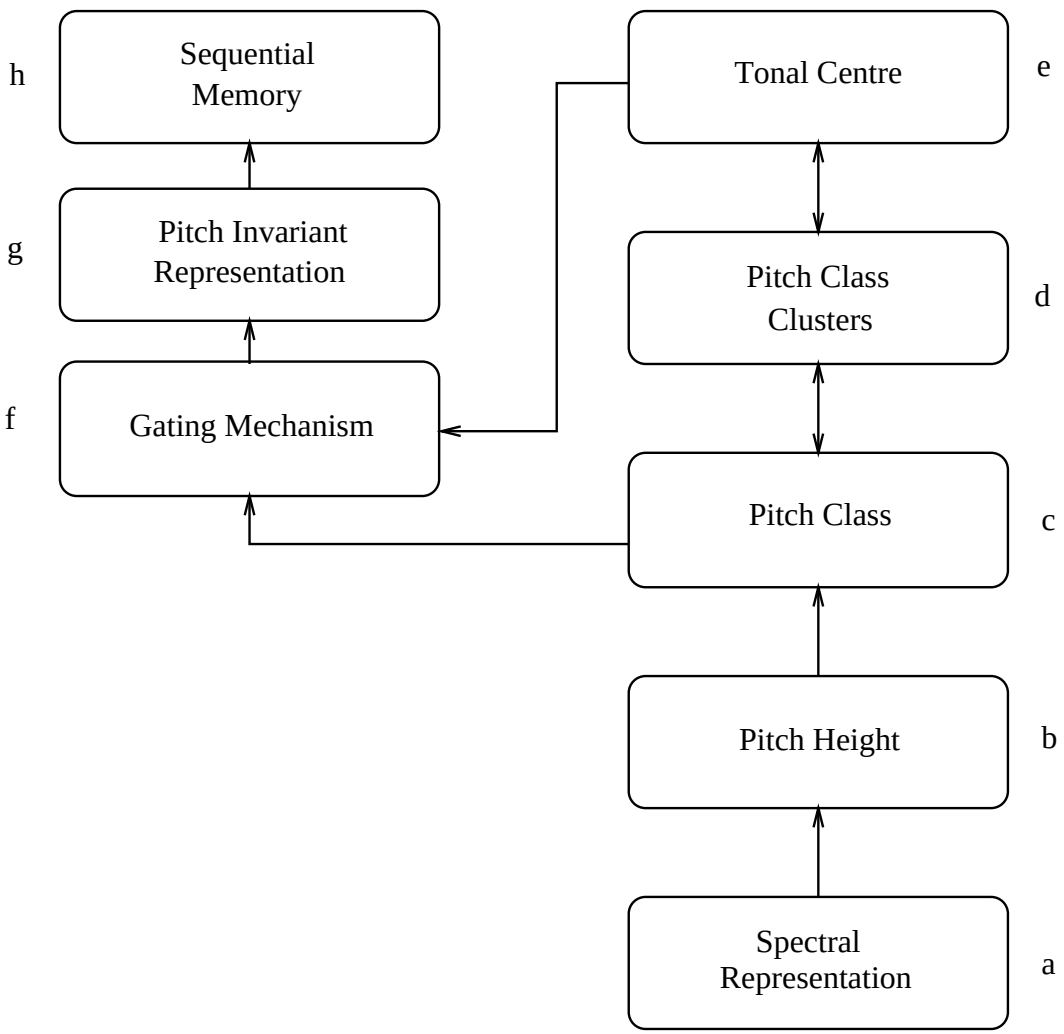


Figure 3.1: An overview of Bharucha's model

lel links connecting the tonotopic levels c and d. Furthermore, Bharucha admits that stability considerations require additional structure (not shown) to ensure that new chord types lead to the recruitment of new higher-level units. These stability issues will be discussed later in the chapter with reference to Adaptive Resonance Theory. A similar learning mechanism is assumed to exist between levels d and e, whereby units in level e become specialized to respond to certain keys. The resulting network, MUSACT, is an interactive-activation network that adjusts its global pattern of activity in a manner consistent with the constraints placed on it by recent input. Its application to the modelling of chord priming is discussed in the previous chapter. Finally the gating mechanism, f, developed in Bharucha [13], consists of a simple arrangement of "sigma-pi" units—learning issues are not considered.

Scarborough, Miller and Jones [138] describe a model of tonality induction that is superficially very similar to MUSACT. Once again the network is divided into three layers containing pitch-class units, chord units and key units respectively. Input is read into the lowest layer, with an activation proportional to the corresponding note's duration, where it combines with a decaying trace resulting from recent input. The activity at this level is then fed forward in a linear fashion through a set of weighted connections to the chord level, and on, in a similar manner, to the key level. The authors note that the linearity of the feedforward process means that an equivalent mapping from notes to keys could be performed by a two-level network. The network is hard-wired to reflect the authors' knowledge of chord and key membership and, unlike MUSACT, has no top-down connections to account for priming effects. In spite of its simplicity, the authors report good results for key induction. Scarborough et al. also propose a gating mechanism for generating the pitch-invariant representation required by a sequential memory.

Griffith [67] uses an **ART2** network (see later) to classify patterns of activity in a "memory vector [that] becomes populated by all the pitch classes that have occurred" during complete presentation of each of a number of simple nursery rhymes. He associates the **ART2** classes with the key of the corresponding melody. The introduction of information relating to the frequency of occurrence of each pitch enables the network to make a better estimate of the melody's key. He goes on to use a Kohonen Feature Map to compare the "hidden unit representations" generated by a "Simple Recurrent Net", a sequential memory trained on the set of (non-pitch-invariant) melodies.

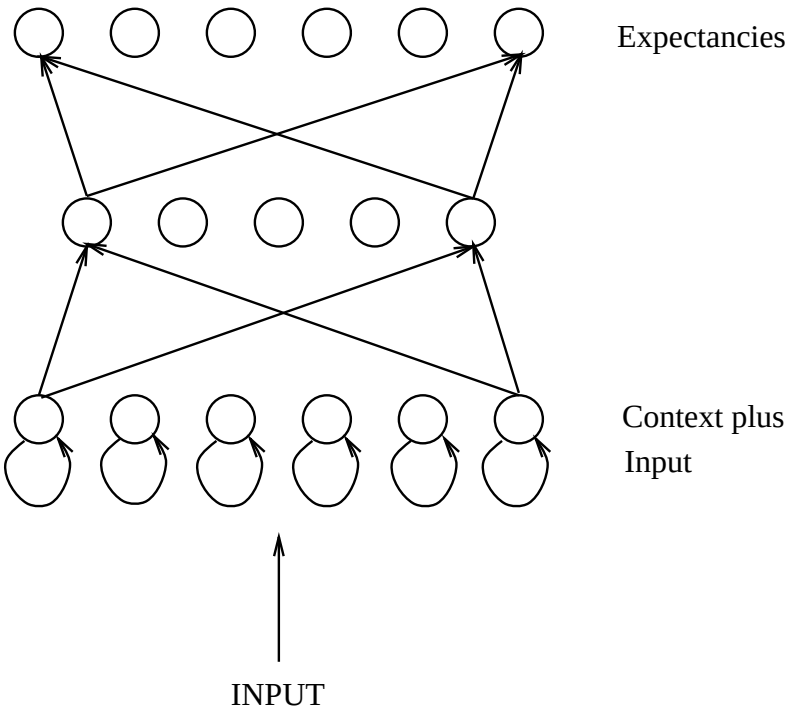


Figure 3.2: The basic Jordan network.

In the next section I shall discuss two models of the sequential memory itself.

3.2.2 Models of memory for musical sequences

Bharucha and Todd's model

Bharucha and Todd [15] propose a model of sequential memory based on the network developed by Jordan [88]. Their model is directed towards the generation of expectancies and they are careful to distinguish between *schematic* expectancies, which they define as "culture-based expectancies for events that typically follow familiar contexts", and *veridical* expectancies, which are "instance-based expectations for events that follow in a particular familiar sequence". In [15] the input consists of sequences of chords, the chords being supplied in a pitch-invariant (functional) representation, whilst Todd [149] employs essentially the same network to process tone sequences. The basic Jordan network is shown in Figure 3.2. The input layer is made up of cells, each of which activates in response to the occurrence of a particular chord (pitch). These input units also possess excitatory recurrent connections, which enable this layer to store a "decaying memory of the sequence presented up to any given point in time"— the time constant of this decay is set by the weights of the recurrent connections. The

cells in the input layer are fully connected to the hidden layer cells, and these in turn are fully connected to the cells in the output layer. The learning of sequential expectations is accomplished as follows: immediately after the presentation of each element of the input sequence, the *next* element of the sequence is placed at the output layer and acts as the target output vector for the current input pattern. The authors use the back-propagation (BP) learning rule (also know as the generalized delta rule), which performs gradient-descent minimization of a specified error measure in weight-space (see e.g. Rumelhart, Hinton and Williams [135]), to train the network to produce appropriate outputs (expectancies) in response to familiar input (sub)sequences. They employ the network described so far in the modelling of the acquisition of schematic expectancies — the network does not learn to reproduce the precise sequences on which it has been trained, but learns to match the "conditional probability distributions of the sequence set". Examples are given of the reaction of the trained network to single chord stimuli.

To model the generation of veridical expectancies, additional network structure is required. This consists of the "additional context" input units shown in Figure 3.3. These units are known as "plan units" in Jordan's formulation and they represent information regarding the "identity" of the sequence currently being presented. A simple scheme, the one used by Bharucha and Todd, is to represent each sequence to be presented by the activation of a single plan unit—this requires the number of plan units to be equal to the number of training sequences. A "localist" representation such as this requires more input units than would a compressed (e.g. binary-coded) representation, but does not lead to learning difficulties resulting from the spurious similarity of plan representations corresponding to different input sequences. The provision of these additional context units serves, for example, to differentiate the early input patterns resulting from different sequences which begin with the same few notes. By training the network on a number of input sequences and by analyzing its subsequent behaviour in response to test sequences containing, in one case, schematically expected transitions, and in the other, unexpected transitions, the authors show that, in spite of the presence of the additional context units, the network is able to "extract" schematic knowledge whilst performing veridical learning.

Todd [149] used essentially the same network, with the addition of one-to-one feedback connections from the output units to the non-plan input units. This modification allows

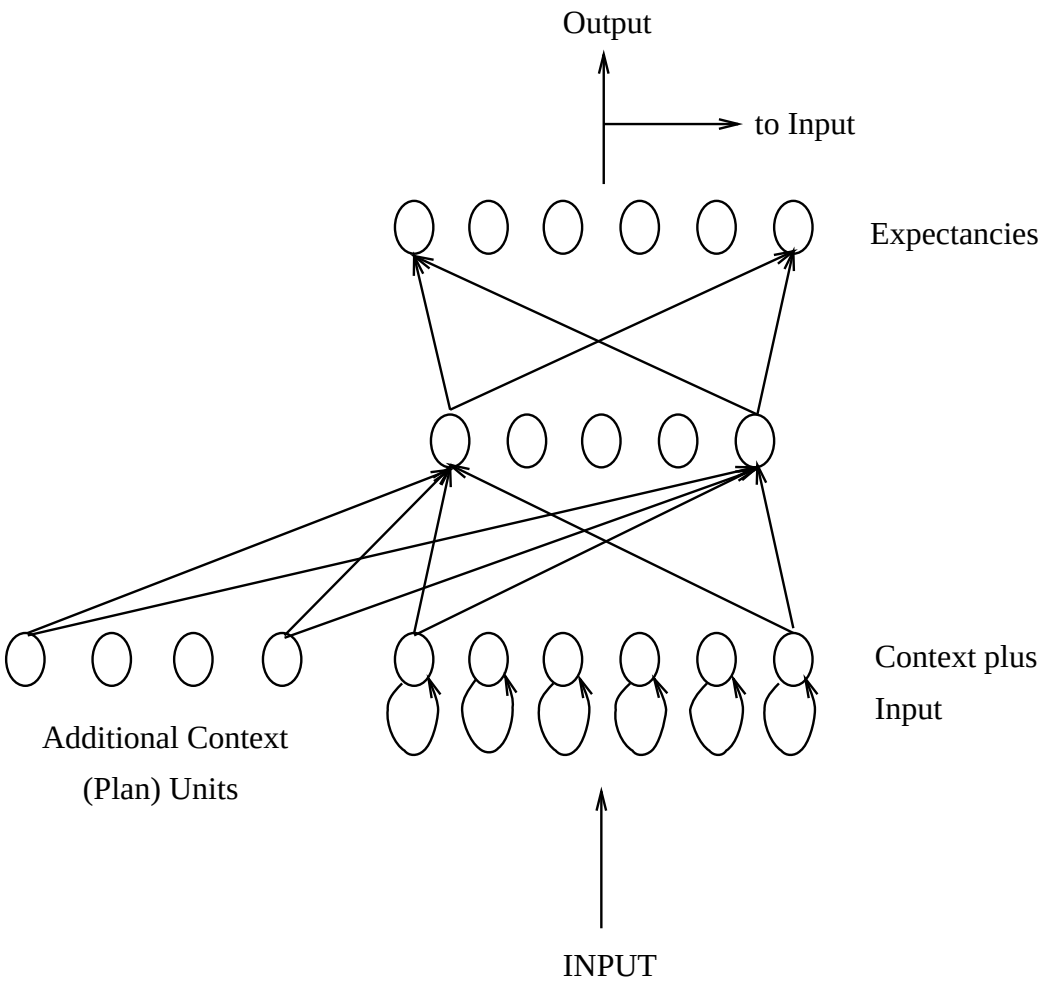


Figure 3.3: The Jordan network with additional context (plan) units.

the network to "step through" a learned sequence in a process akin to melody recall — the output produced at one time-step becomes the input at the next. Since Todd's application was directed towards the memory of sequences representing melodies, he also employed a simple representation of duration, whereby a note was presented at the input layer for a number of "time-slices" equivalent to its duration (actually the inter-onset-interval between its onset and that of the subsequent note). In fact, this means of representing duration requires an additional input unit which activates during the time-slice in which any note first turns on — this modification is necessary to allow the network to differentiate between the sounding of the same note twice in succession and its sounding once but with twice the duration.

In training the network, Todd made a further adjustment to the network's behaviour: whereas during "recall" it is the network's outputs that are fed back to the input in order to build up a context pattern, during training he fed back the training inputs, i.e. the "desired" outputs — this technique, known as "teacher forcing", is one of several aspects of the network's performance which requires "external interference" (see below).

It is worth noting the number of epochs (complete presentations of the training set) required for the network to learn particular sequences: for one sequence of 34 time-steps, the network took 2000 epochs; the learning of two such melodies required 5500 epochs; four required 8500 epochs (N.B. these figures depend crucially on the number of hidden units used, in this case fifteen). In reproducing the melodies, the expected next note was deemed to correspond to that output unit whose activation exceeded a given threshold by the greatest amount. If no output units exceeded the threshold then "the output is interpreted as a rest".

Todd went on to use the trained network in a "composition" task, in which he varied the pattern of activation clamped onto the plan units and seeded the network with a short initiating sequence. The precise details of this process are not so relevant here, but Todd claims that the network performs in ways interestingly different from composition systems based on statistical methods (e.g. Markov chains etc.).

In an addendum to his original article (in [150]), Todd notes that his model is limited in the length of sequences which it can learn, a problem which he attributes to at least two factors: firstly, the network learns each sequence simply as a string of notes "with no higher-level organization to aid in recall"; secondly, with a limited context-span, the network finds sequences such as $ABCDEEEABCDGG$ difficult to learn, since the context patterns

preceding the presentation of the first E and the first G may well be very similar. Todd suggests a hierarchical network as a solution to these problems. His scheme involves the division of the sequence into appropriate chunks (presumably defined by an "expert"), such as the chunks $ABCD$, EEE, and GG. A given network is allowed to learn each of these chunks with a different plan context (e.g. P_1, P_2, P_3 respectively). Subsequently, a "higher-level" network learns the plan sequence (P_1, P_2, P_1, P_3) — the addition of a timing mechanism to coordinate the action of the two networks enables the production of the full sequence.

A critique of Bharucha and Todd's model

Whilst interesting in itself, there are a number of reasons why Bharucha and Todd's model is implausible as a model of of sequential memory:

- The network cannot be considered self-organizing, i.e. able to operate in real-time without external intervention in its behaviour. This is illustrated most obviously by the fact that the training vectors, representing the *next* note in the sequence, are present at the network's output layer, immediately after the occurrence of the *current* note. If the network is assumed to learn sequences by exposure to those sequences in real-time, then, in order for the identity of the next note to be available at the output layer, it must already have occurred. To allow information relating to its predecessor to appear simultaneously at the input layer, it must be delayed in its arrival at that layer until the following note occurs. Working backwards to the start of the sequence, we can see that nothing appears at the input layer until the occurrence of the second event. Not only is this counter-intuitive, and inconsistent with Bharucha's own assertions about the psychological reality of the pitch events themselves, but it also subverts the purpose of the network, namely the generation of expectations. If a note cannot appear at the input layer until the *subsequent* note has already occurred, then any expectations that the first note generates will similarly not manifest themselves until the to-be-predicted note has already occurred. The situation is even worse when one considers a hierarchical memory for a sequence of "chunks" — the occurrence of a chunk cannot be registered at the higher-level input layer until the subsequent chunk has already occurred! The further we go up the hierarchy, the more implausibly long become the time-delays required at the input layer. It is possible to postulate that during the learning of sequences just such

time-delays are operative, but that during some subsequent read-out of expectations they are not, but this requires an external agent to interfere in the behaviour of the network to tell it to stop learning — this point is developed next.

- The network fails to solve the "stability-plasticity dilemma". This balances the requirement for the network to be plastic enough to enable it to continue to learn newly-apparent regularities (in this case new sequences), whilst remaining stable enough to retain previously acquired memories. Suppose that the network has been exposed to a corpus of melodies, which it has learned to reproduce precisely. Now suppose a second, different corpus of melodies (or even a single melody) is presented to the same network for training — assuming that the network is unable to generate correct expectations for these melodies on first hearing, errors will be produced at the output layer which will drive the learning rule to search the weight-space for a new error minimum. Unless the entire set of melodies presented to the network thus far (i.e. both sets of melodies) is presented during the network's search for a new error minimum, there is no guarantee that, once the network has learned the second set of melodies, it will not have "over-written" its memory of the first set. There are two possible solutions: one is to retrain the network on its entire set of learned sequences every time a new sequence is to be learned; the other is to arbitrarily stop all learning at some given point in time, thus sacrificing the ability to learn any more novel sequences. The latter solution essentially trades plasticity for stability — neither solution is particularly appealing as part of a model of sequential memory.

- The third criticism concerns the number of epochs required for the learning of a small number of sequences. As noted earlier, Todd presented each of two sequences to the network over 5000 times before the network was able to learn both successfully. If this network is considered to be a model of sequence learning, it is hard to avoid the conclusion that a similar number of presentations of a simple melody would be required *in vivo*. Todd goes to some lengths to make the presentation of a sequence to the network analagous to the real-life experience, faithfully varying event durations etc. The number of epochs required introduces a considerable disparity in the level of realism. It is not possible to reduce the required number of epochs by increasing the learning rate in the network, since such a measure would likely lead to instability in the gradient-descent

minimization itself, possibly preventing it from locating a sufficiently deep minimum at all. Neither can an appeal to the parallel nature of the network's calculations succeed, since the problem does not lie in the time taken to process each epoch, but in the number of epochs itself.

- The proposed network is able to perform sequence recall but is apparently unable to perform sequence recognition. If we assume that the activation of the plan units denotes the identity of a melody, the lack of any feedback to the plan units implies that there is no way to identify a familiar melody when it is heard. Without such an ability the network will not be guaranteed able to continue the partial presentation of a melody, no matter how familiar. This is clearly at odds with the trivial accomplishment of such tasks in actuality. The problem seems to be connected with the fact that the melody identification information resides in the same field (the input layer) as information relating to the occurrence of pitches. The suggestion that semantic information is involved in a bottom-up process, via a field of cells containing other cells responsive to quite different input, seems rather implausible.

- The fixed weights on the recurrent connections at the input layer make the definition of a note's context rather rigid. Kohonen et al. [95] discuss the advantages of having a context "window" whose size varies with time. This can be seen to be advantageous when considering sequences such as melodies, which are perceived as being structured into phrases or chunks. Briefly, we do not necessarily want the context to extend back across phrase boundaries, i.e. the expectations for continuation of a given phrase should not depend as much upon the notes at the end of the previous phrase as on the same-phrase notes. Such a variable-length context is difficult to implement with fixed recurrent weights.

- The hierarchical arrangement suggested by Todd is rather implausible. It requires that the phrases into which a sequence should best be "parsed" be specified in advance and that the plan-context patterns associated with each of these phrases should be used to train the higher-level network to learn the *phrase sequence*. Originally the plan units were conceived as "identifying" whole sequences corresponding to melodies — Todd's hierarchy requires that each *phrase* have an identifier assigned to it. Furthermore,

it is hard to see how the correct sequence of phrase *plans* can be *extracted* from the presentation of a melody, so that they can be used to train the higher-level network (even allowing for the long time-delay described above).

- The addition of plan units also militates somewhat against the formation of schemas corresponding to prototypical patterns. For instance, suppose the sequence "mi re do" (given in tonic sol-fa to emphasize the pitch-invariant input format) occurs in several different melodies. It seems unreasonable to differentiate between occurrences of this phrase simply because they form part of different melodies — in Bharucha and Todd's network the phrase has strictly to be learned for each different melody in which it appears, not withstanding the fact that this should become progressively easier for each new occurrence, due to the generalization properties of the network.

- The final criticism of Bharucha and Todd's model concerns the nature of the learning algorithm itself. This is an important consideration in the assessment of a scheme which purports to model the equivalent real-life process. The back-propagation (BP) learning algorithm is essentially a mathematical abstraction which enables gradient-descent minimization of error to be performed in a parallel-distributed fashion. The "neuronal" functions which it suggests, for instance the calculation of error derivatives with respect to each incoming weight, have little or no basis in known neural function — Grossberg [70] identifies various ways in which the back-propagation learning algorithm violates "locality" constraints when formally considered as a neural model. This is not to underestimate the usefulness of the BP algorithm in the general application of neural networks to a variety of problem domains — it simply casts considerable doubt on the plausibility of BP as a brain process. As Crick [30] has put it, BP is "far-fetched" as a biological learning mechanism.

Mozer's model

Mozer [113] also applies a recurrent network, trained with the BP learning rule, to the automated "composition" of melodies. He uses a network of the sort studied by Elman [54]. One of the main differences between this network and Jordan's is that the recurrent connections are found at the hidden unit layer rather than at the input layer, enabling context to be built up at this level. In addition, the recurrent connections are not fixed weight but are able to

be learned along with the feedforward weights — Mozer claims that fixed recurrent weights are a weakness of Todd's model. Unfortunately, the added flexibility of Mozer's network is bought at a considerable cost: the BP learning rule applies only to feedforward networks and therefore must be adapted when training networks with recurrent connections. The particular adaptation favoured by Mozer, the "back propagation through time" (BPTT) [135], involves making one copy of the *entire* network for *each item* in the training sequence (presumably this number is known in advance). This drastic measure enables the network to become strictly feedforward, permitting application of the BP algorithm. An alternative learning procedure for recurrent networks, called real-time recurrent learning (RTRL), was introduced by Williams and Zipser [157]. This avoids the necessity to build networks of increasing depth, at the cost of some computational complexity. Since the "real-time" aspect of the RTRL algorithm relies on an approximation that allows weights to change during the presentation of the sequence rather than changes being accumulated until the end of the sequence, the learning rate must be kept small to ensure reliable gradient-descent. The extra complexity required in the training of recurrent connections only serves to call further into question the plausibility of using the BP algorithm in a model of sequence learning.

Mozer also distinguishes his work from Todd's by employing a "psychologically motivated representation of pitch" — he adopts an input representation based on the five-dimensional pitch space proposed by Shepard [142], which, in spite of its complexity, does not encode any information relating to key or tonal centre. He shows that the resulting network is able to generalize from its experience of training sequences, extrapolating those sequences into "unfamiliar" octaves.

Mozer's network suffers from many of the same problems as Bharucha and Todd's, if regarded as a model of sequence learning *in vivo*. In particular, it is not self-organizing and it fails to address the stability-plasticity dilemma. In the next section I shall introduce networks based on Adaptive Resonance Theory (ART).

3.3 Adaptive Resonance Theory and its application

3.3.1 Introduction

This section introduces the principles underlying a family of networks developed by Grossberg et al., under the umbrella of Adaptive Resonance Theory (ART). The introduction is detailed, since many of the ideas are used throughout this thesis. Unlike the supervised networks discussed earlier in the chapter, ART confronts issues of unsupervised learning, self-organization, stability-plasticity and biological plausibility — it therefore forms a solid foundation on which to construct models of human intelligence. In this introduction I shall specifically refer to the **ART2** network, which is capable of classifying, in a stable fashion, analog (as opposed to binary) patterns drawn from an "arbitrary" input environment. This generalizes the **ART1** algorithm which was developed for the classification of binary patterns only. The **ART3** network, which makes some modifications to the **ART2** search mechanism, is introduced in the next chapter. Other variations exist, e.g. **ART2a** [25] and **ARTMAP** [24] (a supervised network which combines unsupervised ART modules), but they all build on the same basic ideas.

3.3.2 The ART2 network

Structure

The **ART2** network was introduced in Carpenter and Grossberg [22]. It consists of two fields of cells, denoted $\mathcal{F}_1$ and $\mathcal{F}_2$, fully connected by two matrices (one "bottom-up", the other "top-down") of weighted connections, also known as *adaptive filters*. Crucial to the understanding of ART networks in general is the idea of a cooperative-competitive on-centre off-surround field. This is a field of cells in which each cell is connected to itself by an excitatory recurrent connection (the on-centre), and connected to all other cells in the field by inhibitory connections (the off-surround). Although the **ART2** network is most often described in an "algorithmic" form, most suitable for programming on a conventional computer, it is important to realize that many aspects of its behaviour are modelled on the properties of competitive-cooperative fields. These properties will be highlighted when relevant in the description that follows.

The **ART2** network is shown schematically in Figure 3.4. As can be seen, the field $\mathcal{F}_1$ is

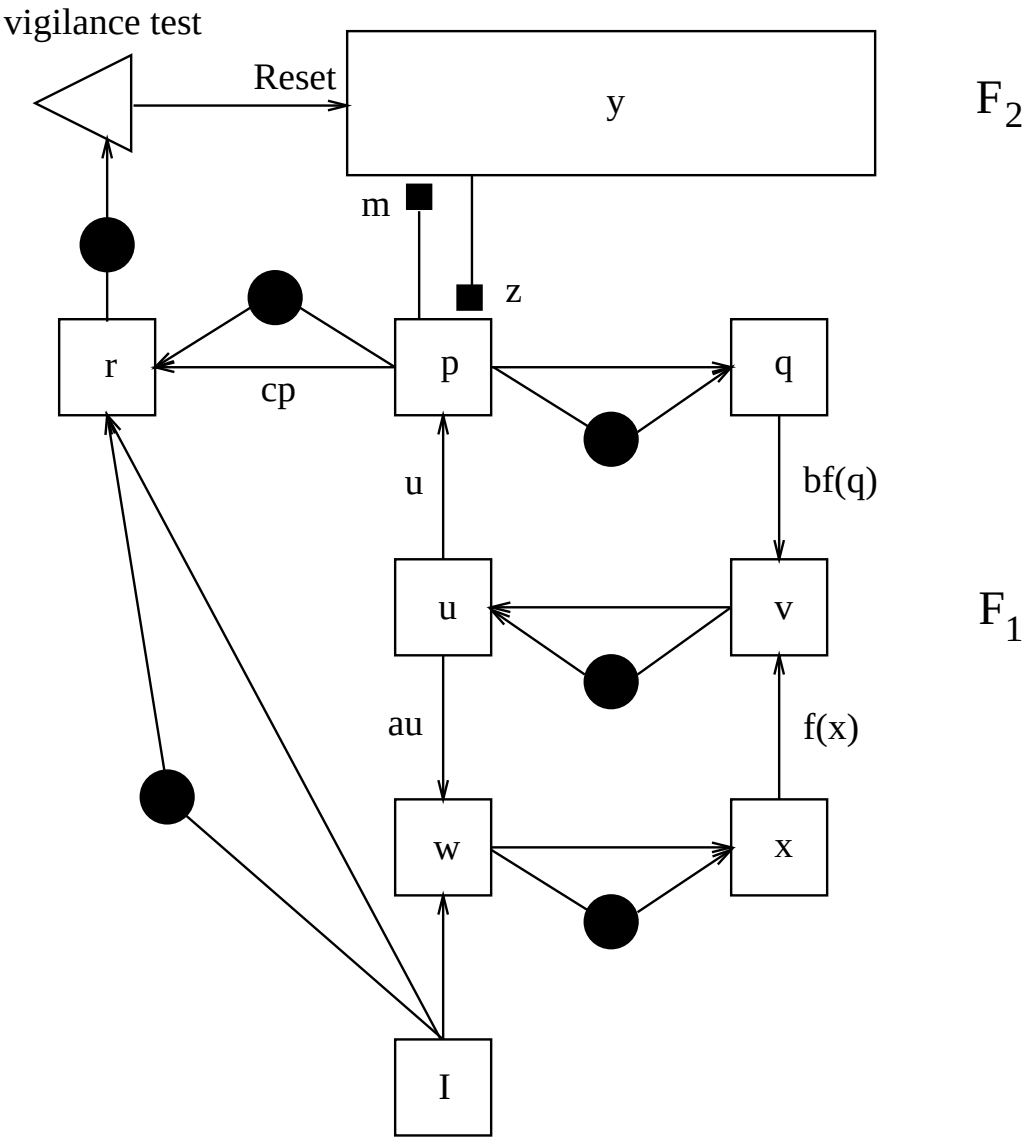


Figure 3.4: The **ART2** network (after Carpenter and Grossberg).

divided into 3 layers. The letters $\mathbf{p}, \mathbf{q}, \mathbf{u}, \mathbf{v}, \mathbf{w}$ and $\mathbf{x}$ denote activity vectors, that is patterns of activity across a subfield of cells. The notation p_i refers to the activation of the i^{th} cell in the relevant subfield. In the description that follows the term "the subfield p" will refer to the set of cells whose activations are described by the vector $\mathbf{p}$. The activation, a_i, of a cell is assumed to vary according to a membrane equation [75] of the form

$$\epsilon \frac{d}{dt} a_i = -Aa_i + (1 - Ba_i)J_i^+ - (C + Da_i)J_i^- \tag{3.1}$$

where ϵ, A, B, C and D are constants and J_i^+ and J_i^- are the excitatory and inhibitory inputs to the i^{th} cell respectively. Variations of this membrane equation will be used extensively in later chapters. In **ART2**, parameters B and C are set to zero and ϵ is set small, implying that the activation a_i proceeds rapidly to its equilibrium value of

$$a_i = \frac{J_i^+}{A + DJ_i^-} \tag{3.2}$$

Thus for the arrangement of $\mathcal{F}_1$ subfields shown in Figure 3.4, in which direct arrows denote one-to-one excitatory inter-field connections and arrows via black discs (after Carpenter and Grossberg) denote the calculation of the length of the incoming vector and the non-specific inhibitory broadcast of this value to all cells of the destination vector, the following equations apply at equilibrium

$$w_i = I_i + au_i \tag{3.3}$$

$$x_i = \frac{w_i}{e + \| \mathbf{w} \|} \tag{3.4}$$

$$v_i = f(x_i) + bfq_i \tag{3.5}$$

$$u_i = \frac{v_i}{e + \| \mathbf{v} \|} \tag{3.6}$$

$$p_i = u_i + \sum_j g(y_j) z_{ji} \tag{3.7}$$

$$q_i = \frac{p_i}{e + \| \mathbf{p} \|} \tag{3.8}$$

where $\| \mathbf{p} \|$ represents the length (L_2 norm) of vector $\mathbf{p}$, where a, b and e are constants and function f is such that

$$f(x) = \begin{cases} 0 & \text{if } x < \theta \\ x & \text{otherwise} \end{cases} \tag{3.9}$$

y_j denotes the activation of the j^{th} cell in $\mathcal{F}_2$, function g implements the "choice" procedure defined below and z_{ji} denotes the weight of the top-down connection between the j^{th} $\mathcal{F}_2$ cell

and the i^{th} cell in the $\mathcal{F}_1$ subfield p. The summed term in Equation 3.7 therefore refers to filtered top-down input from the so-called "category representation field", $\mathcal{F}_2$. It is this top-down input which distinguishes ART networks from other, simpler, competitive-learning models.

$\mathcal{F}_2$ is a competitive-cooperative field configured so as to contrast-enhance the filtered input pattern that it receives from $\mathcal{F}_1$. The simplest arrangement, and that most often used, is one in which the contrast-enhancement is sufficiently fierce to implement "choice" at this level, i.e. the cell with maximal bottom-up input acquires its maximum activity, whilst all other cells in the field have their activity set to zero. In addition, a mechanism is assumed to exist whereby $\mathcal{F}_2$ cells can be "enduringly reset" i.e. have their activation pegged at zero, thus removing them from the competition for activation at that field — the usefulness of such a mechanism is outlined below. The bottom-up input T_j to the j^{th} $\mathcal{F}_2$ cell is given by

$$T_j = \sum_i p_i m_{ij} \tag{3.10}$$

where m_{ij} denotes the weight of the bottom-up connection linking the i^{th} cell in subfield p to the j^{th} $\mathcal{F}_2$ cell.

Learning in the bottom-up and top-down adaptive filters is carried out via the "pseudo-Hebbian" learning equations

$$\frac{d}{dt}m_{ij} = \epsilon g(y_j)(p_i - m_{ij}) \tag{3.11}$$

$$\frac{d}{dt}z_{ji} = \epsilon g(y_j)(p_i - z_{ji}) \tag{3.12}$$

where ϵ now denotes the learning rate. If the function g implements choice, we can see that both the bottom-up and top-down weights abutting the j^{th} $\mathcal{F}_2$ cell tend towards the vector $\mathbf{p}$ at times when the vector $\mathbf{p}$ is such that that cell receives the maximal bottom-up input. At such times no learning occurs at any of the other $\mathcal{F}_2$ cells.

The only remaining part of the network is the so-called "orienting" subfield, r. This receives the input pattern, I, and additional input from subfield p, weighted by a constant, c. Thus at equilibrium

$$r_i = \frac{u_i + cp_i}{e + \parallel \mathbf{u} \parallel + c \parallel \mathbf{p} \parallel} \tag{3.13}$$

The length of $\mathbf{r}$ is compared with a dimensionless parameter known as the vigilance, ρ, the result controlling the delivery of a non-specific reset signal to the cells of $\mathcal{F}_2$.

The next section describes how the network structure described above functions so as to perform stable classification of input patterns.

Function

Before the presentation of any input to the **ART2** network, we assume that all cells have zero activation, the bottom-up weights are set to small, random values and the top-down weights are set to zero. A length-normalized input pattern is presented at the subfield I. Since all cells in $\mathcal{F}_2$ have zero activation, there is no top-down input to $\mathcal{F}_1$, and so the pattern that circulates within this field is solely determined by the input pattern and the threshold linear function f — only supra-threshold input circulates. In fact, we can ensure that the complete pattern $\mathbf{I}$ circulates at $\mathcal{F}_1$ by preprocessing the input. Activation of the subfield p causes input to arrive at $\mathcal{F}_2$ via the randomly initialized bottom-up adaptive filter, and contrast-enhancement rapidly leads to choice at this level. The length of vector $\mathbf{r}$ essentially gives a measure of the degree of match between the vectors $\mathbf{I}$ and $\mathbf{p}$ (it is a function of the cosine of the angle between the two vectors). Since the top-down weights are initially zero, the strong activation of a single $\mathcal{F}_2$ cell has no effect on the pattern circulating at $\mathcal{F}_1$ and thus the degree of match between $\mathbf{I}$ and $\mathbf{p}$ remains maximal (equal to unity). Under conditions in which the degree of match exceeds the value of a parameter known as the vigilance, ρ (where $0 \leq \rho \leq 1$), an adaptive resonance is said to occur. This refers to the stable, complementary activation of both $\mathcal{F}_1$ and $\mathcal{F}_2$.

Only when the network has stabilized into such a state, can significant learning in the bottom-up and top-down adaptive filters proceed. During learning, the bottom-up and top-down weight vectors abutting the active $\mathcal{F}_2$ cell become increasingly "parallel" to the current vector $\mathbf{p}$. In the case of the bottom-up vector this learning will increase the chances of the same $\mathcal{F}_2$ cell activating in response to the same (and similar) input patterns in the future. The top-down weight vector comes to represent a "prototypical" pattern for the activation of the corresponding $\mathcal{F}_2$ cell. After learning, the vector $\mathbf{p}$ becomes an amalgam of bottom-up input from vector $\mathbf{u}$ and the prototypical pattern projected down by the currently active $\mathcal{F}_2$ cell. In the case where this $\mathcal{F}_2$ cell was previously "uncommitted" (i.e. no learning had previously been carried out at that cell), the vector $\mathbf{p}$ remains parallel to $\mathbf{I}$ throughout the presentation of the current input and the "match criterion" remains satisfied, i.e. the degree of match

between **I** and **p** remains greater than or equal to the level of the vigilance.

We now suppose that the network activity is reset to zero and a second input pattern is presented, similar but not identical to the first. Once again the input pattern establishes itself at $\mathcal{F}_1$ and input is fed forward to $\mathcal{F}_2$. We assume that contrast-enhancement of the input leads to the activation of the same $\mathcal{F}_2$ cell as activated in response to the first pattern. This time, however, the amalgam of bottom-up input and prototypical pattern at subfield p will not be perfectly matched with the pattern **I**, due to the slight difference between the first and second input patterns. If the degree of match is sufficient to exceed the vigilance parameter, adaptive resonance and weight-learning will ensue as before. If, however, the degree of match, indicated by the length of vector **r**, falls below ρ, then a reset event is sent to all cells in $\mathcal{F}_2$. This reset event effectively sets the activation of all *active* $\mathcal{F}_2$ cells to zero for the remainder of the presentation of the current input pattern. In the "$\mathcal{F}_2$-choice" case, the single active $\mathcal{F}_2$ cell is reset in this way. Following reset, the competition for activation at $\mathcal{F}_2$ is "re-run", this time without the participation of the "suppressed" former winner, and another cell activates. If this cell is uncommitted then adaptive resonance and learning will follow. Alternatively, if it is committed then adaptive resonance and further learning will only occur if the "vigilance" test is successfully passed. As each subsequent input pattern is presented to the network, the same procedure is followed: the $\mathcal{F}_2$ cell activates, whose weight vector forms the maximum inner-product with the input pattern; the vigilance test is performed and depending on its result either adaptive resonance or reset occurs; in the case of resonance, learning proceeds, in the case of reset, "search" continues; providing $\mathcal{F}_2$ is not "full", search is bound to lead to resonance and subsequent learning.

The activation of an $\mathcal{F}_2$ cell in response to a given pattern is termed "classification". Input patterns are divided into classes or categories indicated by the $\mathcal{F}_2$ cell which they enduringly activate on presentation to the network. The subtlety of the distinctions made between classes is determined by the value of the vigilance parameter during learning: a low vigilance parameter may give rise to given set of input patterns being divided into a small number of broad classes; a high vigilance parameter will result in a larger number of more narrow classes. The classes formed will depend somewhat on the order of presentation of the input patterns and the learning rates used. The number of classes into which input patterns can be placed is limited only by the number of cells in $\mathcal{F}_2$ — if a limit is reached then a sufficiently novel

input pattern remains unclassified in preference to the unlearning of previously established categories.

The **ART2** network performs unsupervised learning of pattern categories — there is no external teacher indicating class membership. It operates in real-time and solves the stability-plasticity dilemma by maintaining the ability to learn at all times, providing that an adaptive resonance is established. The category search process automatically investigates the most "likely" categories first, as indicated by the inner-product of the current input vector with each of the bottom-up weight vectors to $\mathcal{F}_2$ cells. Additionally, **ART2** calculations involve only information available local to the site at which they are performed — this adds to the network's biological plausibility.

3.3.3 ART2 Simulation

This section illustrates the factors which affect the operation an ART network and demonstrates its superiority over the simple Competitive Learning (CL) model. I wrote a program to simulate the operation of the **ART2** network (using the structure shown in Figure 10 of [22]). The simulation was tested on the same input pattern set as that used in the original paper, and identical results were achieved. The writing of an **ART2** simulation highlighted certain aspects of the network's behaviour which are not dealt with in great detail in that paper. In particular, the distinction between slow and fast learning must be made explicit. In the no-learning case, iterative evaluation of Equations 3.3–3.8 results in rapid convergence to equilibrium values. When bottom-up and, more particularly, top-down LTM learning is employed, using a fourth-order Runge-Kutta algorithm to integrate Equations 3.11 and 3.12, there will be a small change in the weights after each iteration, which affects the activities at subfield p, and hence other $\mathcal{F}_1$ activities. Equilibrium at $\mathcal{F}_1$ can be defined as the state in which the magnitude of the activity changes at $\mathcal{F}_1$, resulting from a single iteration, falls below a given, small constant, μ. For small values of μ, equilibrium can require many iterations.

Fast learning can now be defined as a regime in which stringent (μ set very small) equilibrium is achieved in response to the presentation of each input pattern. Note that equilibrium at $\mathcal{F}_1$ implies that the top-down weights have also reached near-equilibrium values. By contrast, slow learning is defined as a regime in which only a small number of iterative evaluations of the STM and weight equations is performed in response to each input pattern,

making equilibrium conditions unlikely. As a result of the specific $\mathcal{F}_1$ dynamics described above, in particular the choice of the parameters a and b as large enough to implement the (weak) "$\frac{2}{3}$-rule" at $\mathcal{F}_1$, the number of iterations required to achieve stringent $\mathcal{F}_1$ equilibrium is typically much greater in cases where an $\mathcal{F}_2$ cell is activated by a pattern that does not match its expectation pattern perfectly, than in cases where a perfect match is obtained or an uncommitted node is activated. This observation forms the basis for the **ART2a** algorithm, which employs "intermediate learning"—**ART2a** sacrifices some of the dynamic properties of **ART2** in favour of algorithmic simplicity and increased speed of execution.

3.3.4 ART2 performance with a chord classification task

Several authors (see earlier) have suggested that chord classification can be carried out by a process involving the simple Competitive Learning (CL) algorithm. The CL algorithm is essentially equivalent to the **ART2** algorithm with the vigilance parameter permanently set to zero (though normalization and noise quenching are not necessarily included). To illustrate the need for the extra stability that results from the ART architecture, the performance of both CL and **ART2** was assessed with reference to an input pattern set representing a variety of three and four-note chords.

The input set is derived from the same set of chords as was used by Leman [104]. It consists of patterns representing three and four-note chords formed by "stacking" major and minor thirds on each of the notes of a one-octave chromatic scale. The mapping onto a one-octave chromatic scale avoids the issue of different chord inversions and, by equating individual pitch classes with scale-note classes, may be thought of as a representation independent of absolute pitch. On each of the twelve pitch-classes is built a major triad (43 %), a minor triad (15%), a diminished triad (4%), an augmented triad (1%), a major seventh chord (1%), a minor seventh chord (1%), a dominant seventh chord (26%), a half-diminished seventh chord (2%), an augmented seventh chord (1%), a minor-with-major-seventh chord (1%) and a diminished seventh chord (7%). The augmented triads and the diminished seventh chords are degenerate, resulting in a total of 115 different chords. The percentages in brackets after each chord type indicate the relative frequency of occurrence of each chord type in music of the classical and romantic period ([104] after [19]) — this information will be used later. Chords are represented by simple 12-dimensional binary patterns: the presence of a given pitch class

in a particular chord is indicated by a "1"; all other pitch classes are set to "0". This binary representation is not that used by Leman, who adds "subharmonic" information in order to facilitate performance of the task to which his network is applied (see below). It is well suited, however, to the current task, in which differentiation between chords is desired.

Fast learning

The input pattern was presented to the **ART2** network under fast learning conditions. The performance of the network was investigated for various values of the vigilance, ρ, including $\rho = 0.0$, which approximates CL, and $\rho = 0.99$, which is high enough to allow distinction between any pair of input patterns. In addition, the input pattern set was presented in five different random orderings. In general the input set is presented repeatedly to the network: the presentation on which "direct access" is achieved (i.e. each pattern activates an $\mathcal{F}_2$ cell without prior reset) and, where appropriate, the presentation on which the weights have fully converged, are both reported.

For a vigilance $\rho = 0.0$, fast learning for five different random orderings of the input set resulted in: 15 categories (2 triads, 13 dyads), 16 categories (1 triad, 15 dyads), 14 categories (1 triad, 13 dyads), 15 categories (1 triad, 14 dyads), 17 categories (2 triads, 15 dyads). In each case, the breakdown given in brackets indicates the nature of the categories formed, as indicated by their bottom-up and top-down weight vectors. For all five trials, direct access is achieved from the second presentation onwards; weight convergence is, by the definition of fast learning, achieved by the end of the first presentation. These results demonstrate that a fast competitive-learning algorithm is not capable of learning to classify the chords in a suitable way — the majority of $\mathcal{F}_2$ cells become specialized in the recognition of dyads. The presentation of a three- or four-note chord will activate these note-pair categories accordingly, possibly leading to "ties" in the competition for activation at $\mathcal{F}_2$ — in general ties are either broken by a random choice, simulating the effect of "noise", or by selecting the $\mathcal{F}_2$ cell (category) with the lowest index. Another means of breaking apparent ties is given below. The final point to note is that the number and nature of the categories formed, is dependent on the order in which the input set is presented: for each different ordering a different categorization was achieved. In this sense, the **ART2** algorithm, at least in its fast-learn manifestation, resembles the "leader algorithm" for unsupervised clustering, in which

new clusters, represented by a prototype (codebook) vector, are added incrementally when no previously defined cluster is capable of "including" a newly-presented input pattern.

When the vigilance is increased to 0.99 (in fact any $\rho > 0.97$), fast learning results in 115 categories for each ordering of the input set, i.e. each chord forms a category of its own. Direct access and weight convergence once again result from a single presentation. This immediately illustrates the advantages to be gained from use of ART's additional network structure.

For intermediate values of the vigilance, different classifications are made—clearly, for any given ordering of the input set, the number of categories formed will be monotone increasing as the vigilance varies between zero and one. An interesting case is that in which the vigilance is set to $\rho \approx 0.965$— this is sufficient to prevent dyads (not used here) from being categorized with superset triads or four-note chords, triads from being categorized with other chords sharing two notes, and four-note chords from being categorized with other four-note notes sharing three notes. This level of vigilance will, however, permit triads to be categorized together with superset four-note chords (i.e. four-note chords with which they share three notes) and vice versa. The use of this vigilance level allows us to investigate whether categories can be formed corresponding to the triads alone, so that four-note chords are classified in terms of their subset triads. This corresponds with the idea that the four-note chords are variations of their root triads, e.g. that F major seventh can be thought of as a variation of F major — chord usage often supports such an interpretation, extended chords being introduced at points where their root triads would perform a similar tonal function.

For a vigilance of 0.965, fast learning for the five different random orderings of the input set resulted in: 41 categories (40 triads, 1 four-note chord), 40 categories (all triads), 41 categories (40 triads, 1 four-note chord), 40 categories (all triads) and 41 categories (40 triads, 1 four-note chord). Direct access and weight convergence were achieved as before. These results show that reduction to triad categories cannot be guaranteed under fast-learning conditions—depending on the ordering of the input set, at least one four-note category can remain.

In spite of its speed and efficiency, the fast learning condition does not provide a realistic model of chord learning in vivo. In addition comparisons with the competitive-learning algorithm under such a regime are perhaps unfair, since CL is not usually used in this way. The next section illustrates comparative performance under slow learning conditions.

Slow learning

As detailed above, under slow learning conditions the number of iterative calculations of the system equations performed on presentation of each input pattern is restricted so that the weights evolve slowly in response to repeated presentation of the input set. Once again we look at the classifications performed by **ART2** network for various values of vigilance, including that corresponding to simple competitive learning.

For a vigilance of 0.99, slow learning resulted in 115 categories for each of the five different random orderings of the input set, i.e. each chord forms a category of its own. The category structure evolves slowly — some $\mathcal{F}_2$ cells begin to learn "amalgam" patterns but these become incompatible with the high vigilance once a certain degree of learning of the top-down weights has been performed. Using the same convergence criteria as in the fast-learn experiments, direct access was achieved on presentation 399, 239, 321, 233, 298. These numbers depend on the precise choice of learning rate (number of iterations per pattern). The number of presentations before weight convergence varies between the $\mathcal{F}_2$ cells. For the first input ordering, the earliest weight convergence was after 720 presentations, though some weights had not fully converged after 4500 presentations. Under slow-learning conditions, weight convergence is not guaranteed (except for the limiting case where $\rho = 1$), and the top-down weight vector corresponding to a given category can "move around" in a neighbourhood whose maximum size is determined by parameter settings, in particular the settings of ρ, θ, a, b and the learning rate.

For a vigilance $\rho = 0.0$, slow learning for the five different random orderings of the input set resulted in: 19 categories (1 four-note, 11 triads, 7 dyads), 16 categories (1 four-note, 5 triads, 10 dyads), 16 categories (1 four-note, 6 triads, 9 dyads) and 15 categories (8 triads, 7 dyads). Clearly competitive learning, without the additional constraints imposed by an ART architecture, is not capable of forming a category structure that corresponds to our idea of clearly differentiable chords, even under conditions of very slow learning. Additionally, if the make-up of the input set were to change, there would be no guarantee that what had previously been learned would not be "washed away" by subsequent exposure.

For the five different orderings of the input set, an intermediate value for the vigilance, viz. $\rho = 0.965$, yielded: 45 categories (31 triads, 14 four-note), 46 categories (31 triads, 15 four-note), 39 categories (31 triads, 8 four-note), 45 categories (36 triads, 9 four-note) and

48 categories (37 triads, 11 four-note). Direct access was achieved on presentations 180, 115, 272, 123, and 317. Thus, even under slow-learning conditions, the category structure that emerges for intermediate values of the vigilance depends on the ordering of the input set. We cannot, therefore, reliably extract categories corresponding to the triads alone from the repeated presentation of a single random ordering of the input set.

Slow learning with an altered input set

An altered input set was produced which reflected the various frequencies of occurrence of the chords in the input set, as reflected by the figures given above. This consisted of 1153 chords, assembled by including each chord a number of times corresponding to the frequency of occurrence of its chord type, e.g. the C Major chord was included 43 times, the F minor 15 times etc.. Once again five random orderings of the altered input set were produced. Although this input set was generated with reference to statistics taken from music of a specific type and period, it nonetheless permitted an investigation of the effect of a variation in the frequency of occurrence of various chords.

A preliminary study with the slow learning rate used above and the vigilance set at 0.99 showed that the **ART2** network was able to place each chord in a separate category but that direct access took over 200 presentations of the total input set — this is equivalent to over 200 000 individual pattern presentations. This figure represents the number of presentations of the input set required before *all* patterns access their category without reset — some patterns only occur once in the 1053-pattern set, so the number of *patterns* that are presented before direct access is only large so as to allow the most infrequent patterns to be learned sufficiently. In order to make simulations more tractable (and possibly more realistic), the learning rate was adjusted so that direct access was achieved in about 30 presentations of the complete input set. Under these conditions, setting $\rho = 0.99$ categorized all chords separately regardless of the ordering of the input set. Setting $\rho = 0$ once again failed to produce a "satisfactory" categorization.

For a vigilance of 0.965, repeated presentation of the full pattern set resulted in categories corresponding to the 40 triads alone, for three of the five random orderings of the input set. For one of the other orderings, there were 41 categories corresponding to the 39 triads plus 2 dominant seventh chords (the remaining triad, a diminished triad, was categorized with one

of the dominant seventh chords). For the remaining ordering of the input set, there were 38 categories: 2 categories corresponding to dominant seventh chords (each subsuming a diminished triad), 35 triad categories, and 1 category corresponding to a diminished seventh chord, which subsumes the remaining 3 (diminished) triads. These results suggest that we cannot reliably expect the network to extract triad categories alone under the current conditions — the most frequently presented triads, viz. major and minor triads *always* form categories, but the least frequently presented triads, in particular diminished triads, can be subsumed into categories corresponding to the more frequently-presented four-note chords, dominant sevenths and diminished sevenths.

Slow learning with a "graded" input set

A final set of slow-learning trials demonstrated that triad categories can be exclusively formed, for the intermediate vigilance case ($\rho = 0.965$), if the patterns to which a network is exposed in the early part of its learning history are predominantly triadic, extended chords being initially infrequent but becoming increasingly frequent with time. To illustrate this point input sets were generated from the same chords used previously, but with the relative chord frequencies changing to reflect the progression described above: the first 1000 chords consisted of 90% triads and 10% four-note chords (relative occurrence-frequencies being preserved within these groups); the second 1000 chords consisted of 80% triads and 20% four-note chords; the third 1000 chords consisted of 70% triads and 30% four-note chords; the subsequent distribution was that used earlier (approximately 60% triads). Five input sets were generated, each with random pattern ordering constrained by the desired distribution. For each input set, slow-learning led to direct access after approximately 7500 pattern presentations and always resulted in a category structure corresponding to the 40 input triads. The success of this "seeding" of the network's memory with triadic patterns relies somewhat on the fast-code-slow-recode property of ART learning that is exploited in the **ART2a** algorithm.

Summary

This simple chord-classification task illustrates many of the features of ART networks. The simple competitive-learning algorithm is not capable of performing a satisfactory classification under fast, slow or intermediate-learning conditions. High vigilance **ART2** can form a sepa-

rate category for each chord for any learning condition. Intermediate values of the vigilance can be contrived so as to produce a desired categorization, although the statistics and the ordering of the input set can affect the results as shown.

The simulations described above are unrealistic models of chord learning in many ways. Chords are not experienced in random orders and the learning rate and vigilance parameters do not necessarily remain constant throughout exposure. Genuine supervision may be available, as in the explicit teaching of chord names, in which case a supervised-learning network (such as **ARTMAP**) may be more appropriate.

The idea that chords might be recognized from an input set whose elements are already in an absolute-frequency invariant format (e.g. scale-notes), allows us to avoid having to classify each type of chord at a large number of absolute frequencies, thus avoiding the rather inelegant "yoking" solution proposed by Bharucha (see Section 3.2.1). The absence of any "absolute" chord categories does imply, however, that the corresponding chord units do not play a rôle in the key-identification procedure. This again is incompatible with Bharucha's model, but is compatible with the models of [138] and [67]. If absolute chord units do not exist then the priming effects elicited from the MUSACT model must be explained in other ways. The previous chapter suggested that the priming effects resulted from the learning of typical sequences of non-absolute (functionally defined) chords. In this way, chord priming would occur at a stage after key identification. Priming also provides another mechanism for "breaking ties" when ambiguous chords are presented, i.e. the context can play a part in the identification of the function of a given chord in a sequence. Later chapters investigate how typical chord sequences might be learned.

3.3.5 L' ART pour l'art

Gjerdingen [63] [62] [64] used the **ART2** network in a system which he dubbed "L' ART pour l'art" (ART for art's sake). The input field consisted of cells, each of which was assumed to respond to one of "34 specific, low-level musical features". These features included melodic scale-degrees, bass scale-degrees, "flags" for melodic/bass contour and even contrapuntal dissonance. Bharucha [14] criticizes the specificity of some of these features and Gjerdingen himself is careful to make no claim that the particular features chosen constitute the basic elements of music perception. When one of these features is detected, the relevant input unit

activates and inhibits somewhat the activation of cells "still responding to tones from the recent past". $\mathcal{F}_1$ thus stores a decaying trace of recent events, in addition to the current input; for this reason $\mathcal{F}_1$ is referred to as a short-term memory (STM). Gjerdingen adjusts the $\mathcal{F}_1$ parameters so that the "span" of STM is about 4 to 6 events. The more permanent coding of information in the weighted connections between fields is termed long-term memory (LTM).

"ART pour l'art" was exposed to six of Mozart's earliest compositions. After 12 epochs, the network had formed a stable category structure in response to the $\mathcal{F}_1$ patterns resulting from presentation of the input sequences. Gjerdingen restricted the number of $\mathcal{F}_2$ cells used, i.e. the number of categories, to 25, presumably by adjusting the level of the vigilance parameter accordingly. In spite of this restriction, Gjerdingen reports encouraging results regarding the network's ability to achieve "a musically sound categorization of much of the [input] domain". An interesting point to note is that Gjerdingen was able to improve the network's performance (as assessed by an expert observer) by introducing simple metrical information. He achieved this by modulating the "attention" paid to input events with reference to their metrical position — certain events, particularly those occurring on strong beats, were "marked for consciousness". Similar ideas will be developed in Chapter 8.

Gjerdingen ends by introducing the idea of a hierarchy of **ART2** modules, whereby "categories" activated at $\mathcal{F}_2$, feed activation forward to a second STM field, $\mathcal{F}_3$. The evolving pattern of activation at $\mathcal{F}_3$ is then itself categorized at a further field, $\mathcal{F}_4$. In this respect, fields $\mathcal{F}_3$ and $\mathcal{F}_4$ are more like the sequential memories described earlier, in that $\mathcal{F}_4$ is intended to classify sequences of *single items*, represented as a pattern of activity across $\mathcal{F}_3$. This functionality differs in a subtle way from that of fields $\mathcal{F}_1$ and $\mathcal{F}_2$. The distributed patterns of activity at $\mathcal{F}_1$ result from the pooling of information regarding a variety of different events that have occurred in the recent past; each $\mathcal{F}_2$ cell learns to respond to a variety of broadly similar patterns rather than to a pattern specifying a precise sequence of input events. Gjerdingen notes this distinction and surmises that a more complex formulation is required for the levels $\mathcal{F}_3$ and $\mathcal{F}_4$. The precise nature of this formulation, involving the so-called "masking field", and its application to the processing of item sequences, is the subject of the next chapter.

3.4 Miscellaneous

3.4.1 More Applications of Neural Networks in Music

For completeness, this section briefly describes a number of applications of neural networks in music not detailed above.

Leman [104] describes the application of a Kohonen Feature Map (KFM) to the classification of chords. The KFM is a network that uses unsupervised learning to achieve a reduced-dimension topological mapping, i.e. similar input patterns are classified by "similar" nodes. The similarity of classifier nodes is usually measured by the distance separating them on a two-dimensional "grid" of such nodes. Leman shows that, with a particular representation of chords (incorporating Terhardt's subharmonic information), certain structures, notably the "circle of fifths", emerge at the output layer.

Desain and Honing [34] introduce a model of the "quantization" of musical time, which is implemented using a type of interactive-activation network. The function of each cell is relatively complex, however, and the model does not correspond with any standard connectionist model.

Katz [92] [91] presents a model of musical pleasure based on a connectionist framework. He posits that both harmonic and melodic sequences will be pleasurable if there is a smooth, coherent transfer between adjacent elements. He measures the coherence of a given transfer by monitoring resultant "boosts" in the total activation of portions of his network.

Bellgard and Tsang [7] use a network of Boltzmann machines in a task involving chorale harmonization. The Boltzmann machine is similar to a Hopfield net, with the addition of a simulated annealing process to favour the discovery of a *global* minimum in the network's "energy space" — training the network involves excavating minima in energy space. The authors show how an extended Boltzmann machine, imbued with some additional stylistic constraints, is able to generate plausible harmonizations when presented with a single melodic part. The network design allows global constraints, as opposed to local constraints, to dominate its productions.

Large, Palmer and Pollack [102] use a Recursive Auto-Associative Memory (RAAM) model, which uses a three-layer auto-associative feed-forward network, trained with the BP learning algorithm, to generate compressed representations of sections of simple melodies, and,

from these to reconstruct "facsimile" sections. The training was deliberately curtailed before "memorization" of the input melodies was achieved. The relative structural importance of a melodic event was then related to the fidelity with which it was able to be reconstructed. The values so obtained corresponded well with those derived from an associated improvisation task and separately from Lerdahl and Jackendoff's theory of time-span reduction [105].

Linster [107] has applied a standard multi-layer perceptron to the analysis of grouping in rhythmic structures.

3.4.2 Models of Sequential Memory

Various models of sequential memory have been proposed in the literature, though many have not yet been applied to the modelling of music perception. The majority of these models employ networks similar to those of Jordan [88] and Elman [54], in which a standard supervised learning algorithm (BP, BPTT or RTRL) is used to train a multilayer network with recurrent connections. For instance, Port [124] uses Elman's simple recurrent network in a dynamic model of linguistic sequence processing. Mozer [114] has developed a general taxonomy which includes such models, classifying them with reference to the precise way in which they implement short-term memory. Mozer's taxonomy also covers models based on input from tapped delay lines, such as the Time Delay Neural Network (TDNN) [153] [101]. These networks, once again using BP to perform supervised learning, have been successful in recognizing phonemes and isolated words from input spectrograms, though they require a considerable amount of task-specific tuning.

Fahlman [57] demonstrates that the recurrent architectures described above can be trained using alternative learning algorithms. He uses his Cascade Correlation (CC) learning algorithm to train a modified Elman network (the modifications are necessitated by CC's ability to increase incrementally the size of the hidden layer) and shows certain advantages, in particular faster learning and an ability to train incrementally. Incremental learning results from the fact that all input weights to a given hidden unit are frozen when that unit is added to the network — the stability-plasticity dilemma is still not fully addressed, however, since the output weights from all hidden units remain plastic indefinitely.

Schmidhuber [139] describes the design of a "neural sequence chunker" which, although it uses gradient-descent-based supervised learning, incorporates concepts similar to those devel-

oped in later chapters. Schmidhuber's networks are based on his idea of "history compression". He allows a recurrent network to attempt to learn a given sequence, so that each event learns to predict the next. All those inputs which are subsequently poorly predicted by the network are passed as input to a higher level network, which in turn learns to predict those events only. Thus a hierarchy of recurrent networks is formed, whereby the low-level network learns "local" sequence information and the higher-level network learns more temporally extended structure. In fact, Schmidhuber describes how the internal representations formed by the higher-level network can be fed back to the low-level network, enabling it to learn the extended structure itself — in this way the hierarchy is "collapsed".

Reiss and Taylor [129] describe a model of temporal sequence storage, which employs "leaky integrator neurons" to implement short-term storage of recent input patterns. A time-delay on the input, implemented using a two-channel gating mechanism, allows the network to learn sequence transitions via a delta rule applied to a single matrix of weighted connections. Unfortunately this input delay makes it difficult for the network to generate *expectations* for forthcoming events (see Section 3.2.2). In addition, there is no mechanism suggested for either sequence classification or extraction of higher-level structure (chunking).

Houghton [77] describes a "competitive queuing model" of sequence (word) learning and recall, which is similar in some ways to the models developed in this thesis. In particular, he rejects "models of the multi-layer perceptron type" as being "unlikely", and develops his model with reference to a number of operational constraints — all of these constraints are satisfied by the models developed later. His network is able to learn short sequences (representing words) bordered by well-defined "temporal edges", i.e. the beginning and the end of words are flagged by the activation of word onset and word offset nodes. The network is not self-organizing, since a *specific* pair of word-nodes must be selectively activated during the learning of a particular word. In addition, his model requires supervision during its "practice" phase. Houghton makes two observations concerning the behaviour of the network during recall: firstly, that "all sequence elements are activated in parallel, but with a graded activation pattern which can be maintained over time"; and secondly, "that elements become gradually more activated until they are the most highly activated, after which their activation is rapidly suppressed". These observations, which Houghton suggests are compatible with experimental evidence, are equally true for the models developed later.

Kosko [96] discusses (Adaptive) Bidirectional Associated Memories (BAM) which are able to store pairwise pattern associations. He shows that, by storing associations between successive patterns in a sequence, the network can support limit-cycle behaviour which can be construed as storage of the entire sequence. A similar idea is proposed by Kleinfeld [93], who uses a Hopfield-type network to generate and "recognize" pattern sequences — learning issues are not discussed.

Wang and Arbib [154] postulate an oscillatory model of STM and use it in a system that performs supervised learning of temporal sequences. A high-level node is trained to respond exclusively *following* the presentation of a given sequence. Sequence reproduction is also discussed.

Chappell and Taylor [27] present a modification of the Kohonen self-organizing map, which allows an "understanding of temporality". The resulting "Temporal Kohonen Map" is analyzed and its ability differentially to classify a set of simple short sequences is demonstrated.

Chapter 4

The Masking Field and the MASKART network

4.1 Introduction

The previous chapter introduced Adaptive Resonance Theory and the **ART2** network. This network was able to classify, in a stable fashion, patterns drawn from an arbitrary input environment. Inputs consisted of patterns of activation across a field of cells—such patterns will be referred to as "spatial patterns". In the **ART2** algorithm the dynamics of the classifying field, $\mathcal{F}_2$, are most often chosen to be "winner-takes-all", so that on each trial the network "chooses" the single most appropriate category. In this chapter I will introduce a modified version of $\mathcal{F}_2$ called a "masking field" [29] whose dynamics are considerably more complex. This complexity makes such fields suitable for applications involving the context-sensitive "parsing" of sequences. Firstly it will be necessary to describe how a spatial pattern of activation is capable of representing temporal order information relating to a stimulus consisting of a sequence of items or events.

4.2 Representing Temporal Order with Spatial Patterns

Grossberg [68] introduces a model of immediate recall of sequences from short-term memory (STM). He rejects previous models based on concepts culled from digital computer technology,

such as binary codes, serial buffers etc., as being inconsistent with serial recall data for human subjects. These data indicate a "bow" in graphs plotting order and probability of recall versus list position. He shows that, by allowing the storage of an item in STM to be represented by the non-zero activation of the cell that responds to that item, item-order information can be stored by the relative activation of the relevant cells. For instance, suppose the activation of a cell that responds to a particular event e_n is designated x_n. In response to the sequence (e_1, e_2, e_3, e_4) temporal order information will be stored correctly in STM if the cell activations are such that $x_1 > x_2 > x_3 > x_4$. Grossberg postulates that the *performance* of an item sequence such as this from STM is achieved by subjecting the cells to "non-specific arousal" which boosts, in parallel, the activations of all cells with non-zero initial activation; when the activation of a cell reaches a threshold value it activates its corresponding "motor code" (which embodies the sequence of operations needed to perform the item) and immediately quenches its own activity to zero. During non-specific arousal, therefore, the cell with the highest activation will reach the threshold first and the corresponding event will be performed, followed by each of the events in order of decreasing cell activation. Thus a sequence of items can be correctly performed from STM if the relative cell activations correspond to the item order in the manner indicated. This performance mechanism also precludes the alternative representation of order whereby the sequence above would be represented by $x_1 < x_2 < x_3 < x_4$.

Grossberg further constrains the activation pattern by introducing the "Principle of Code Invariance" which says that when a cell becomes active in STM, in response to the occurrence of a given item, the activations of previously active cells should only change by a single multiplicative factor. The rationale given for adopting this principle is that it makes the coding in STM compatible with the invariant coding of patterns in long-term memory (LTM). Thus if the activation of the i^{th} cell, immediately after it has activated in STM, is denoted u_i, and the activation of this cell in STM causes other cell activations to be multiplied by w_i, then the pattern of activation will evolve as in Table 4.1. A second constraint, "The Normalization Rule", states that the total STM activation grows from an initial value of u_i to a finite maximum M. Grossberg characterizes this rule with the equation

$$S_i = u_1\phi_{i-1} + M(1 - \phi_{i-1}) \tag{4.1}$$

where S_i is the total STM activation after i items have been presented and ϕ_i is the so-called load parameter that is a measure of the "inhibitory design" of the STM field. The constraints

Time	x_1	x_2	x_3	x_4
t_1	u_1	0	0	0
t_2	u_1w_2	u_2	0	0
t_3	$u_1w_2w_3$	u_2w_3	u_3	0
t_4	$u_1w_2w_3w_4$	$u_2w_3w_4$	u_3w_4	u_4

Table 4.1: The evolution of STM activations on presentation of the sequence (1,2,3,4) (after Grossberg (1986))

on ϕ_i are that $\phi_0 = 1$ and $\lim_{i\to\infty} \phi_i = 0$. A simple choice, which characterizes a "homogeneous field", would be one where $\phi_i = \theta^i$, where $0 < \theta < 1$.

In cases where the Invariance Principle and the Normalization Rule both hold, the shunting parameters can be specified thus

$$w_k = \frac{u_1\theta^{k-1} + M(1-\theta^{k-1}) - u_k}{u_1\theta^{k-2} + M(1-\theta^{k-2})} \tag{4.2}$$

in which u_i represents the activation of the cell representing the i^{th} item when it first activates in STM, M and θ are defined as above and $k > 1$. If all $u_i = u_1$ then it is shown that a bow appears in STM at the J^{th} item such that J is the maximal j such that

$$(u_1^{-1}M - 1)(1-\theta)\theta^{j-2} > 1 \tag{4.3}$$

I have written a short program to show how the position of the bow varies with a range of parameter choices. Grossberg accordingly suggests a definition of the Transient Memory Span of STM, this being the length of the longest list that can be represented accurately (i.e. without bowing) in STM. This measure differs from the more familiar Immediate Memory Span ([111]), which can reflect interference from LTM. Grossberg suggests that the two values are constrained by the condition $TMS < IMS$ and proposes that plausible values might be $TMS = 4$, $IMS = 7$.

When this model of STM is incorporated into an adaptive resonance network it will be necessary to make slight changes to the formulation described above. However, the important result of this work, viz. that temporal order information can be represented by relative cell activation in STM, will be used throughout this thesis. A field of cells which represents

information in this way will be referred to as a "working memory".

4.3 The Temporal Chunking Problem

Having established how information pertaining to item sequences can be represented in a working memory, it is now necessary to outline the demands that such a representation makes on a higher-level field (henceforth referred to as $\mathcal{F}_2$) connected to the working memory (referred to as $\mathcal{F}_0$ for consistency with later models) by an adaptive filter. $\mathcal{F}_2$ seeks to classify or parse the item sequence into a series of sub-sequences that is in some sense "optimal". Many of the ideas reviewed here were originally developed with reference to the parsing of the phoneme sequences that make up spoken language. It is my belief, however, that such ideas can usefully be applied to the issues concerning the parsing of musical sequences—that is one of the aims of this thesis.

The basic idea of this section is that the $\mathcal{F}_2$ cells representing those sub-sequences that together constitute an optimal parsing of the sequence represented at $\mathcal{F}_0$, should activate most strongly. The word "optimal" may need some clarification—Grossberg [69] regards an optimal parsing as one that is "predictive", as defined in the following passage:

> " ...How does $\mathcal{F}_4$ [my $\mathcal{F}_2$] know how to automatically group, or parse, the total event list represented across $\mathcal{F}_3$ [my $\mathcal{F}_0$] into sub-lists that have the best a priori chance of predicting the future within the context defined by the unique past represented by that list."

It is interesting to note in this passage the explicit link that is made between an optimal parsing and the enhanced ability to "predict the future". Those organisms that are able to anticipate the likely continuations of environmental sequences and to process expected events more efficiently by being thus primed, will be at an advantage in a world that exhibits some regularity. Such an idea is reminiscent of the work on musical parsing and musical expectation (priming) cited in Chapter 2 and Chapter 3.

Grossberg refers to the parsing task described above as the "temporal chunking problem". The solution which he proposes employs a version of $\mathcal{F}_2$ known as a masking field. The behaviour of this field is constrained so that, a priori, cells in $\mathcal{F}_2$ that respond to broader regions of $\mathcal{F}_0$, i.e. longer sequences (up to some maximum length), should be able to *mask*

(suppress) the activity of cells that respond to less broad regions (shorter sequences). This constraint is consistent with the observation that a sequence is not necessarily best parsed into the individual items out of which it is composed, in spite of the fact that those individual items are bound to be at least as familiar (in terms of frequency of occurrence) as the lists which they form. The masking constraint thus introduces a bias to perceive a sequence (up to a given maximum length) as a *chunk* rather than as a series of single items. It will be seen later how this bias can be overridden by prior learning but for the moment it will suffice to show how this so-called "sequence masking principle" is realized by the dynamics of $\mathcal{F}_2$.

To be consistent with the sequence masking principle Grossberg postulates that cells in $\mathcal{F}_2$ should be connected only to a subset of the $\mathcal{F}_0$ cells and, accordingly, that $\mathcal{F}_2$ should be divided into various sub-fields, a sub-field corresponding to the set of $\mathcal{F}_2$ cells connected to the same *number* of $\mathcal{F}_0$ cells. A further principle is employed, namely the "Principle of Self-similarity", which states that an $\mathcal{F}_2$ cell should possess a set of "masking parameters" which reflects the number of $\mathcal{F}_0$ cells to which it is connected. In this way $\mathcal{F}_2$ cells which are connected to broader regions of $\mathcal{F}_0$ (i.e. a greater number of $\mathcal{F}_0$ cells) have a built-in competitive advantage in STM, as required by the sequence masking principle. To avoid such cells exhibiting this advantage in inappropriate situations, the masking field is constrained so that, in the absence of a learned bias, an $\mathcal{F}_2$ cell is only able to activate if all of the $\mathcal{F}_0$ cells to which it is connected are active.

Grossberg [69] summarizes the main properties of a masking field with four rules whose importance justifies their inclusion here:

> "The main properties to be achieved are all tacitly stated in the sequence masking principle. They may be broken down as follows:
>
> 1. List representation—The unordered sets of items in all realizable item lists, up to a maximal list length, are initially represented in the masking field.
> 2. Masking parameters increase with list length—The masking parameters of masking field nodes increase with the length of item lists that activate them. This rule holds until an optimal list length is reached.
> 3. Masking hierarchy—A node that is activated by a given item list can mask nodes that are activated by sublists of this list.

4. List selectivity—If a node's trigger list has length n, it cannot be supraliminally activated by lists of length significantly less than n."

It is important to note that these rules apply to the behaviour desired of a masking field *before* significant learning has taken place in response to a particular environment—they are principles derived to give the masking field an *a priori* bias towards the chunking of input sequences. These principles are not intended, however, to "second guess" the regularities of a particular environment. The issue of learning will be dealt with later. First I shall describe the particular way in which the masking field model, described by Cohen and Grossberg [29] embodies the principles detailed above.

4.4 The Masking Field

Cohen and Grossberg [29] describe computer simulations of a masking field that exhibits the properties described above. The differential equations which fully describe the dynamic behaviour of the masking field are complex but they can be analyzed so as to reveal the way in which they correspond with the ideas previously outlined. The equations are as follows

$$\begin{aligned}\frac{d}{dt}x_i^{(J)} &= -Ax_i^{(J)} + (B - x_i^{(J)})\left\{\sum_{j\in J} I_j\left[\frac{1}{|J|}(1-p_{|J|}) + r_{ji}^{(J)}p_{|J|}\right]z_{ji}^{(J)} + D|J|f(x_i^{(J)})\right\} \\ &\quad -F(x_i^{(J)}+C)\frac{\sum_{m^{(K)}} g(x_m^{(K)})|K|(1+|K\cap J|)}{\sum_{m^{(K)}}|K|(1+|K\cap J|)} \qquad (4.4)\end{aligned}$$

$$\frac{d}{dt}z_{ji}^{(J)} = \epsilon f(x_i^{(J)})(LI_j - z_{ji}^{(J)}) \qquad (4.5)$$

The first equation describes the way in which the activation of the i_{th} $\mathcal{F}_2$ cell, x_i, changes over time. It can be broken down into three parts thus

$$\frac{d}{dt}x_i = -Ax_i + (B - x_i)P_i - (C + x_i)Q_i \qquad (4.6)$$

where A represents a decay constant, B and $-C$ represent the cell's maximum and minimum activation levels respectively, P_i represents the total excitatory input to the i^{th} $\mathcal{F}_2$ cell and Q_i represents its total inhibitory input.

P_i can be written as the sum of two factors: the input activation, gated by LTM weights, from the set J of $\mathcal{F}_0$ cells to which the $\mathcal{F}_2$ cell is connected; and a self-excitation term which varies with the cell's size (defined below).

$$P_i = \sum_{j \in J} I_j p_{ji} z_{ji} + D|J| f(x_i) \tag{4.7}$$

where f is the sigmoid function

$$f(x) = \frac{(\max(x,0))^2}{f_0 + (\max(x,0))^2} \tag{4.8}$$

Q_i consists of the sum of inhibition from all cells in $\mathcal{F}_2$

$$Q_i = \sum_m g(x_m) E_{mi} \tag{4.9}$$

where E_{mi} is the strength of the inhibitory connection from the m^{th} $\mathcal{F}_2$ cell to the i^{th} $\mathcal{F}_2$ cell and g is the sigmoid function

$$g(x) = \frac{(\max(x,0))^2}{g_0 + (\max(x,0))^2} \tag{4.10}$$

The principle of self-similarity is implemented by allowing some of the parameter choices for a given $\mathcal{F}_2$ cell to rely on the "size" of that cell. The size of a cell is defined as the number of $\mathcal{F}_0$ cells to which it is connected. In addition, the total excitatory connection strength to an $\mathcal{F}_2$ cell is constrained to be equal to 1, regardless of the number of $\mathcal{F}_0$ cells to which it is connected. Cohen and Grossberg present a physical rationale for this constraint: they postulate that the physical volume of a cell is related to the number of connections incoming from $\mathcal{F}_0$ but that this increased volume serves to dilute the effect of each input. Thus the total effective input connection strength remains constant irrespective of cell size. This constraint is characterized by the equation

$$\sum_{j \in J} p_{ji} = 1 \tag{4.11}$$

Furthermore, there is some random variation incorporated into each of the connection strengths p_{ji} (this random element is found in the term $r_{ji}^{(J)}$ in Equation 4.4), so that each of them differs slightly from $\frac{1}{|J|}$. As noted above, the degree of self-excitation also varies with cell size so that larger cells have a competitive advantage, other things being equal.

Similar constraints govern the strength of the inhibitory connections between $\mathcal{F}_2$ cells. The inhibitory connection, E_{mi}, from the m^{th} $\mathcal{F}_2$ cell, which is connected to the set of $\mathcal{F}_0$ cells K, to the i^{th} $\mathcal{F}_2$ cell, connected to set J, depends on the magnitude of set K and the magnitude

of the intersection of sets K and J. The rationale for the latter assumption is that $\mathcal{F}_2$ cells that respond to similar input cells will be located "close" to each other in $\mathcal{F}_2$ and therefore the lateral inhibition between them will be strong. Once again the total inhibitory connection to a given $\mathcal{F}_2$ cell is normalized to unity so that

$$\sum_m E_{mi} = constant = F \tag{4.12}$$

Cohen and Grossberg show in more detail how the differential equation describing the activation of masking field cells (Equation 4.4) is built up.

The second differential equation (Equation 4.5) describes the change in the LTM traces over time. This equation therefore embodies the capacity of the network to learn. It falls within a class of pseudo-Hebbian associative learning laws widely used in self-organizing neural networks. Its basic assumption is that during learning the LTM traces abutting the i^{th} $\mathcal{F}_2$ cell should tend towards a constant multiple (L) of the activity of the $\mathcal{F}_0$ cell that drives them. To prevent inappropriate learning this tendency is gated by the term $f(x_i)$ so that learning only occurs at active $\mathcal{F}_2$ cells. At small values of x_i the sigmoid function is faster-than-linear, which allows learning to enhance any initial advantage in STM enjoyed by a given cell. The effect of learning will be shown in the results of the computer simulations described later in this chapter.

4.5 Computer Simulations of a Masking Field

Whilst the differential equations given above fully describe the behaviour of a masking field, a full appreciation of this behaviour can best be gained by computer simulations of the masking field dynamics— some results from such simulations are presented in [29]. I decided to repeat these simulations by writing a program based on the contents of their paper. A number of factors influenced this decision. Firstly, the intention was to use the masking field in modelling the perception of musical sequences and it was clear that this would necessitate a number of extensions to the formulation described by Cohen and Grossberg. It therefore seemed reasonable to begin by attempting to reproduce the behaviour exhibited by their formulation before adapting it to the particular task required. Secondly, unlike many neural network models, the masking field is essentially dynamic—this is emphasized by the fact that its behaviour

can be described solely in terms of a differential equation involving the time derivative of the cell activations. The dynamic nature of the model presents problems when one attempts to describe on paper the response of a masking field, with a particular set of parameters, to a given input. In presenting the results of their simulations Cohen and Grossberg concentrate on showing the equilibrium state of a masking field under various conditions, although they do (p. 1882) give one example of how the masking field proceeds to equilibrium. It was felt necessary, therefore, to carry out simulations in which the dynamic behaviour of the masking field could be made explicit by animating a graphical display of the cell activations as they change prior to equilibrium. It was further hoped to be able to encapsulate the essence of these simulations in 3-D plots (cell versus activation versus time), the success of which can be judged by referring to Figures 4.1-4.6 etc.. A third reason for wishing to repeat Cohen and Grossberg's results was to provide independent confirmation of those results. I am unaware of the publication of any such work to date and the simulations described in the first part of this chapter are thus useful in that respect. Indeed, the simulations described here, and the questions that they raised, have served to clarify some ambiguities and errors in Cohen and Grossberg's original paper.

In the light of the discussion above the aims of my initial simulations were four-fold: to write a program that could be easily adapted to the simulation of masking fields of various sizes; to generate an animated graphical display of the evolving pattern of activation and to summarize this information in the form of 3-D plots; to make the specification of the masking field structure and parameters as easy as possible for the user; and to ensure that the program ran at a speed which allowed realistic exploration of the masking field's behaviour. Before describing aspects of this program it is necessary to define various terms used

- the *masking field* will be known as $\mathcal{F}_2$ and will be considered as consisting of an array of cells each having a level of activation, x, at a given time. As before, each $\mathcal{F}_2$ cell is connected to a set of $\mathcal{F}_0$ cells, the magnitude of this set determining the *size* of the $\mathcal{F}_2$ cell.

- a *population* of cells is defined as a group of $\mathcal{F}_2$ cells connected to precisely the same set of $\mathcal{F}_0$ cells (N.B. Cohen and Grossberg use the term population in a slightly different way).

- a *section* is defined as a group of $\mathcal{F}_2$ cells that are connected to the same number of $\mathcal{F}_0$ cells. Thus all cells in a given population are in the same section but the converse is not true. By definition, all cells in the same section have the same size. The terms "the first section" or "section 1" refer to the group of all size-1 cells in $\mathcal{F}_2$ etc. .

In writing the program an attempt was made to allow the structure of the masking field to be specified in the simplest way possible. It was realized that much of the information pertaining to the structure of the masking field (e.g. the size of the $\mathcal{F}_2$ cells, the lateral inhibitory weights, the initial LTM connection strengths etc.) could be derived from a matrix which simply specified the connection pattern between $\mathcal{F}_0$ cells and $\mathcal{F}_2$ cells. Since all cells in a given population are connected to the same set of $\mathcal{F}_0$ cells, this description was further reduced to a matrix describing the connection pattern between $\mathcal{F}_0$ cells and $\mathcal{F}_2$ populations. Although this information was sufficient to specify a masking field, more information was added which, though technically redundant, facilitated checking for errors etc.. This information defined the number of cells in $\mathcal{F}_0$, the number of cells and populations in $\mathcal{F}_2$, the extent of the various sections of $\mathcal{F}_2$, and the number of cells per population in each of these sections. The setting of the last of these values is of interest. For a given population it indicates the maximum number of different time orderings of the $\mathcal{F}_0$ items to which members of that population are connected, which can *each* be represented by a *distinct* masking field cell. For full coverage of all possible time orderings the number of cells per population would be $S!$ where S is the section to which the population belongs. In Cohen and Grossberg's network, however, the number of cells per population is equal to 4 for all populations. A related number is the number of populations per section. For full coverage of all possible groupings, this number should equal ${}^{N_0}_{S}C$, where N_0 equals the number of cells in $\mathcal{F}_0$. Thus the maximum number of $\mathcal{F}_2$ cells required in each section is $\frac{N_0!}{(N_0-S)!}$. For meaningful results it is necessary to use at least three sections (1,2 and 3) in $\mathcal{F}_2$ (as do Cohen and Grossberg) or, preferably, four sections. Grossberg [68] proposes that the optimal (maximal) chunk size is approximately equal to the Transient Memory Span of STM, which is assumed throughout this thesis to be 4. The total number of $\mathcal{F}_2$ cells required for full coverage of all possible time ordered patterns consisting of four items or less is thus

$$N_2 = \sum_{S=1}^{S=4} \frac{N_0!}{(N_0 - S)!} \tag{4.13}$$

The fact that this number increases very rapidly as $\mathcal{F}_0$ grows constitutes an important problem for the masking field model, a problem which will be addressed in the next chapter.

The program allows the user to specify information in two places: the first of these is a section at the start of the program where the majority of the information relating to the network structure is defined at compile-time; the second is a data file which is read into the program at run-time—this contains information relating to the masking field parameters as well as the matrix detailing the connectivity between $\mathcal{F}_0$ and $\mathcal{F}_2$. The rationale for splitting this information in this way is a follows. The information that is defined at compile-time is semi-permanent i.e. it relates to the structure of a network, and as such is unlikely to change between trials; allowing it to be specified at compile-time simplifies the program by avoiding the need for a lot of dynamic memory allocation. By contrast the parameter information contained in the data file is likely to be changed frequently—having it loaded in at run-time allows the user to change the parameters without having to re-compile the program.

The program allows the user to load in a previously saved pair of data files corresponding to LTM traces (weights). In the absence of these data files the LTM matrices are initialized such that the LTM connection strengths (p_{ji}) are consistent with Equation 4.11 and the "LTM Densities" (z_{ij}) are set to unity. Next the program generates the $(N_2 \times N_2)$ matrix representing the $\mathcal{F}_2$ inhibitory weights and the vector of N_2 integers representing the sum of inhibitory weights to each $\mathcal{F}_2$ cell. The user is then asked to load in a pattern of $\mathcal{F}_0$ activations as input, under the influence of which the masking field is allowed to proceed towards equilibrium. The integration of the differential equation describing the change in $\mathcal{F}_2$ activations is performed using a 4th Order Runge-Kutta algorithm with fixed step-size (Press et al. [128])—the step size was set small enough to ensure stability. A number of measures were taken to improve the speed of the integration process. In particular it was realized that the total inhibitory input was the same for each cell in a given population and therefore that this value need only be evaluated for each population rather than for each $\mathcal{F}_2$ cell.

The hybrid C/Objective-C program runs on a NeXTstation computer, with a Motorola 68040 processor running at 25 MHz. A typical time to reach equilibrium is approximately 45 seconds. The simple animated graphical display was designed using the Interface Builder program for the NeXTStep GUI.

4.6 The results of the initial simulations

Initial results did not agree with those published by Cohen and Grossberg. It was discovered that this was due, in part, to an error in the parameter values printed—this error was corrected after consultation with the authors. In addition, it was discovered that, in order for the masking field to behave as described, it was necessary for each cell to send some inhibition *to itself*. This is not made clear in the original paper and is unusual in the light of previous work relating to cooperative-competitive fields—it should be noted, however, that the *net* self-effect is still excitatory for the range of parameters given.

Once these corrections were made the simulations agreed with those of Cohen and Grossberg to within 0.1%. This level of agreement was surprisingly good bearing in mind the inclusion of a random element in the initialization of the values p_{ji}. The results are shown in Figures 4.1-4.6. Each Figure shows histograms representing the $\mathcal{F}_0$ (input) vector, the gated input to $\mathcal{F}_2$ (masking field) and the equilibrium $\mathcal{F}_2$ activations. To the right of each of these histograms is a number representing the value of the highest activation in that histogram. Above the histograms is a 3-D plot of the activations of the $\mathcal{F}_2$ cells as they change between the initial presentation of the input and the equilibrium state; in each case the most active cell at equilibrium is labelled with the set of $\mathcal{F}_0$ cells to which it is connected. For the sake of clarity the 3-D plots only show cell activations greater than zero; the fact that some cells become negatively activated (hyperpolarized) is obvious from the histogram of equilibrium activities. The omission of negative activations from the 3-D plots does not decrease the usefulness of such plots since hyperpolarized cells play no part in the competition for activation, the process which the 3-D plots are intended to portray.

In each case the $\mathcal{F}_2$ cell with the highest equilibrium activation is one of the cells connected to exactly the same set as the set of active $\mathcal{F}_0$ cells. To achieve this result it can be seen that $\mathcal{F}_2$ cells which are connected to larger subsets of $\mathcal{F}_0$ can mask the activity of other smaller cells in spite of receiving less bottom-up input than those cells. This behaviour is only exhibited, however, when all of the $\mathcal{F}_0$ cells connected to the larger $\mathcal{F}_2$ cell are active. This is consistent with the third and fourth restrictions on masking field behaviour listed in Section 4.3. In Figures 4.1-4.3 the masking field parameters are chosen such that only one $\mathcal{F}_2$ cell remains active at equilibrium, whereas in Figures 4.4-4.6 the cooperative-competitive balance is altered so that a more "distributed" pattern of $\mathcal{F}_2$ activation is achieved at equilibrium. In

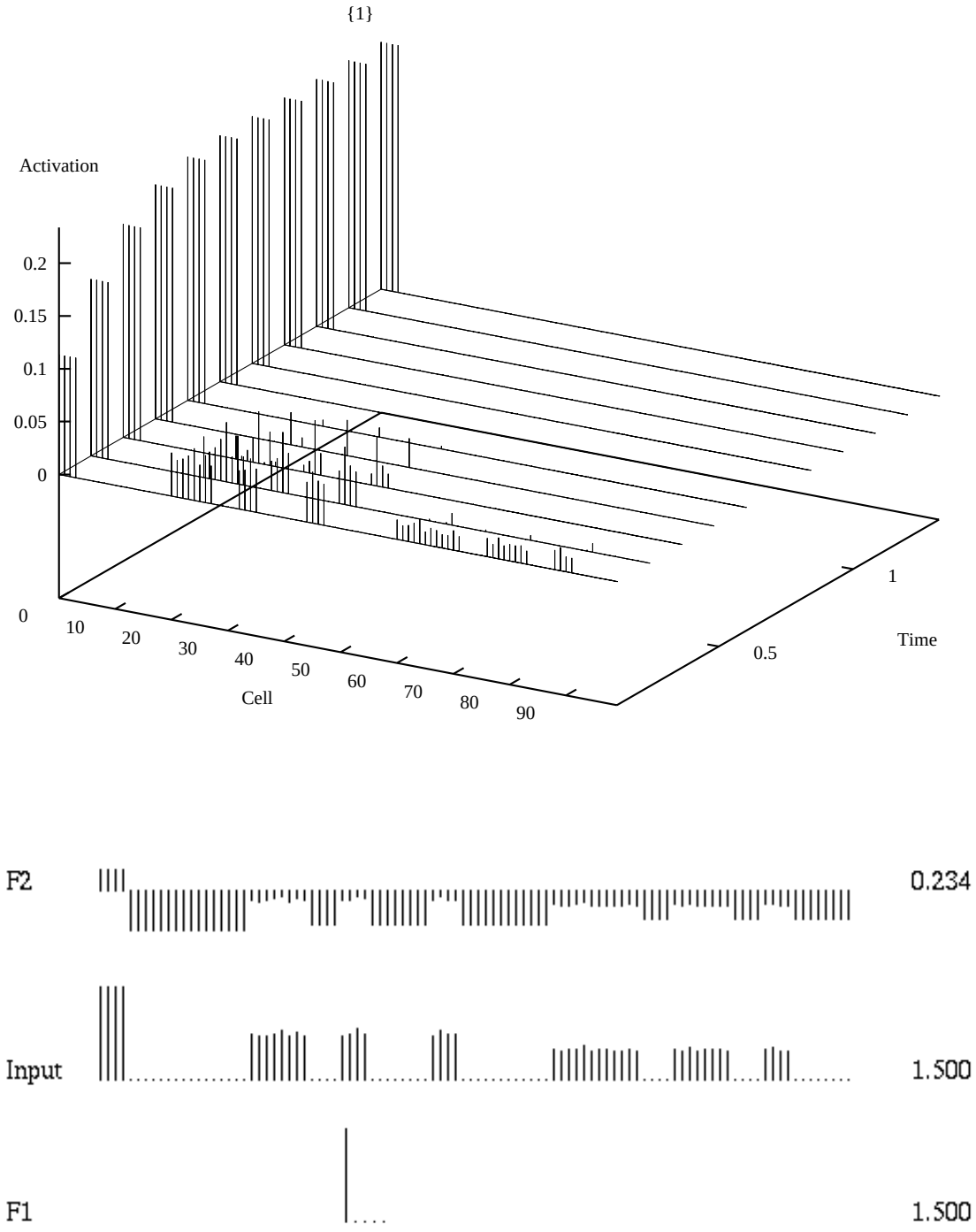


Figure 4.1: The masking field's response to the sequence (1) for $C = 1$, $F = 1088$, $f_0 = 0.16$, $g_0 = 1$.

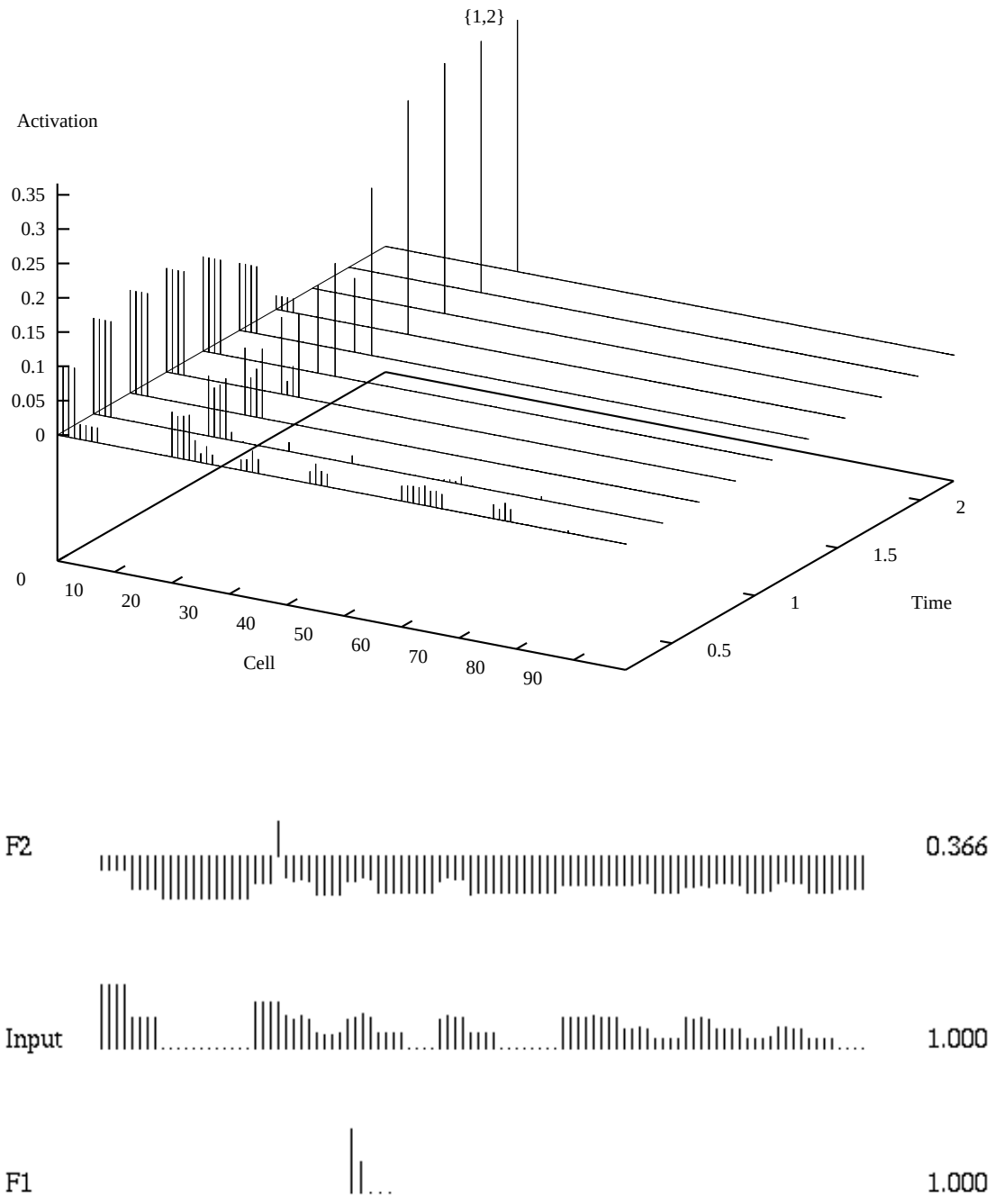


Figure 4.2: The masking field's response to the sequence $(1, 2)$ for $C = 1$, $F = 1088$, $f_0 = 0.16$, $g_0 = 1$.

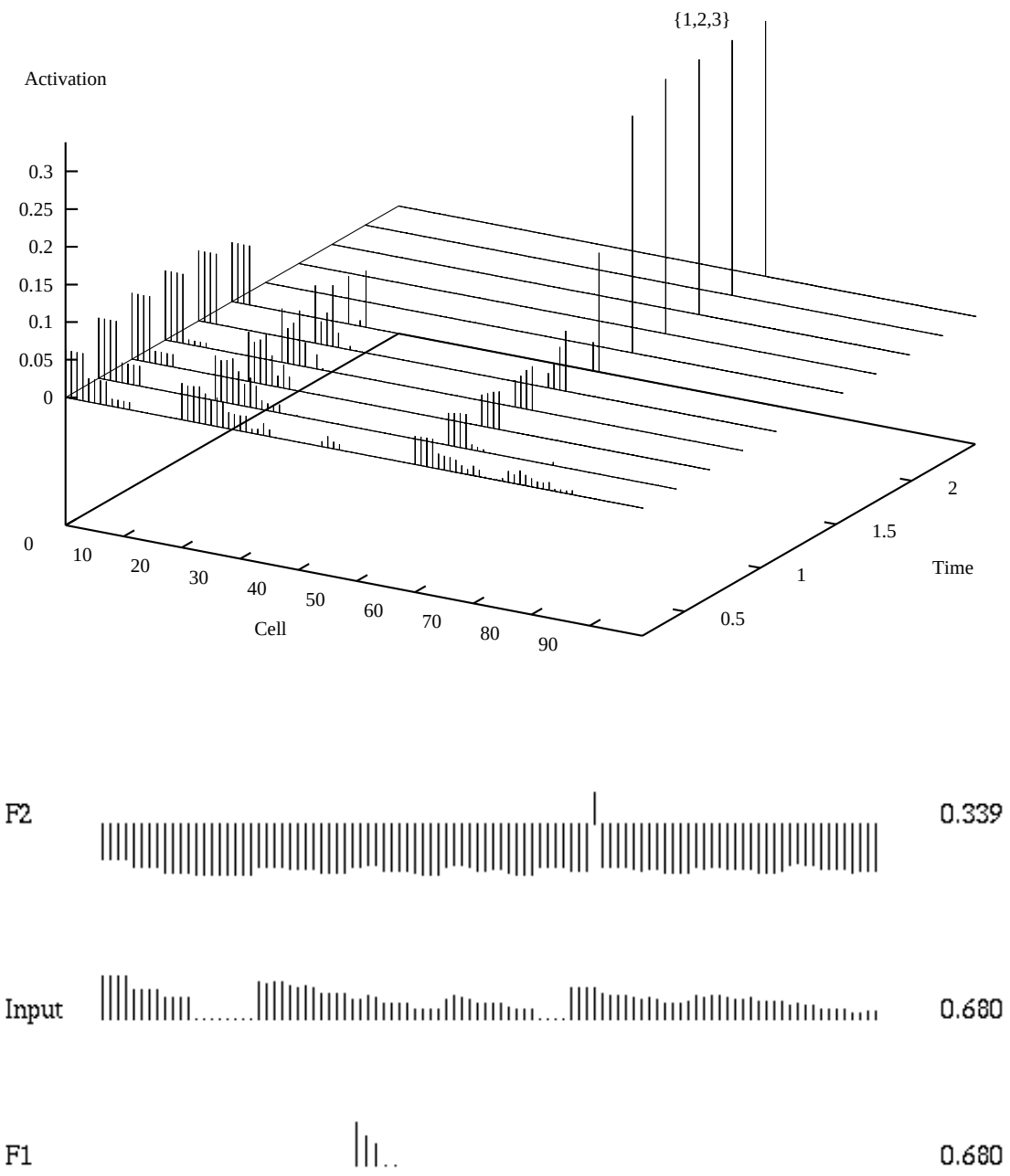


Figure 4.3: The masking field's response to the sequence $(1, 2, 3)$ for $C = 1$, $F = 1088$, $f_0 = 0.16$, $g_0 = 1$.

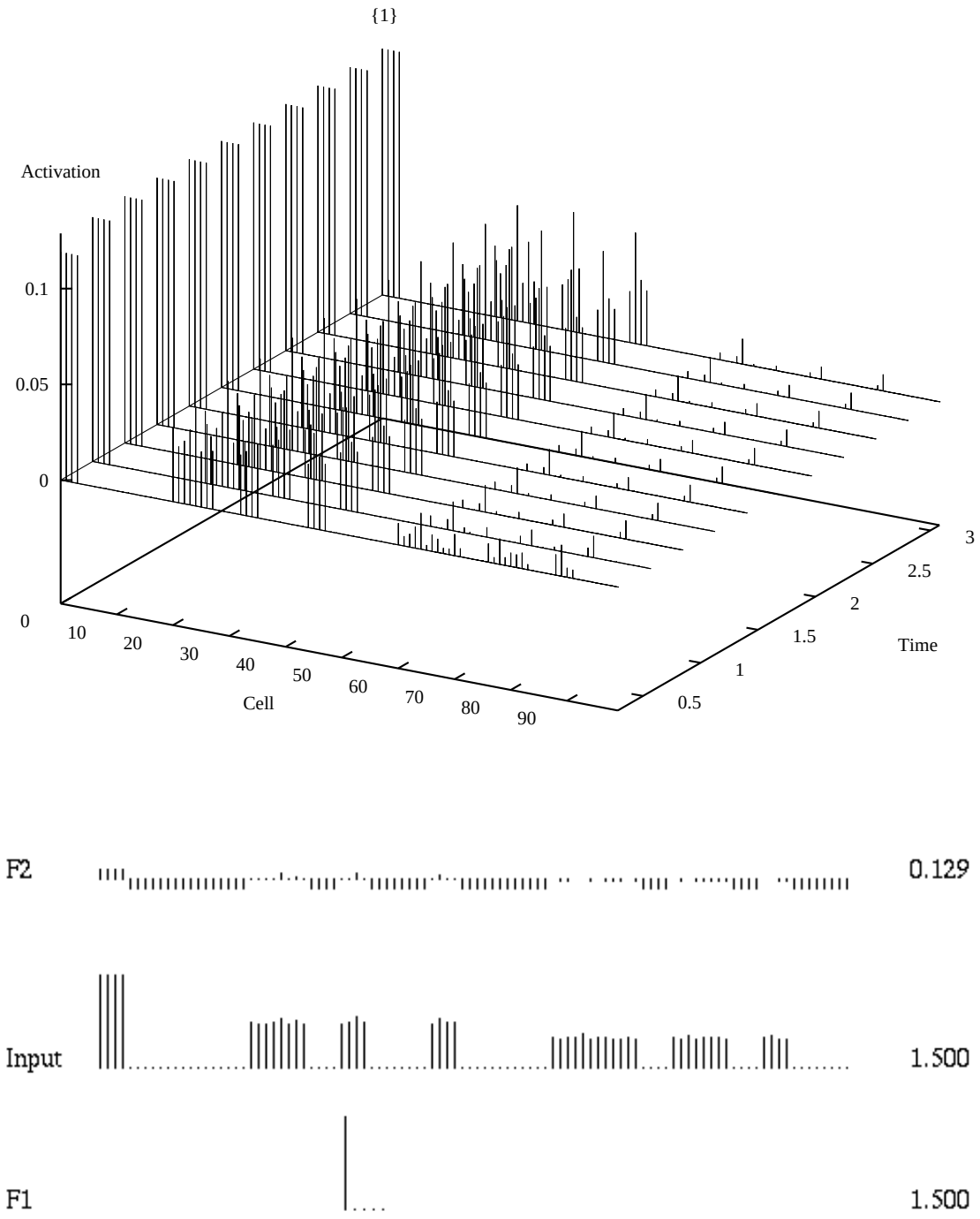


Figure 4.4: The masking field's response to the sequence (1) for $C = 0.125$, $F = 8704$, $f_0 = 0.16$, $g_0 = 1$.

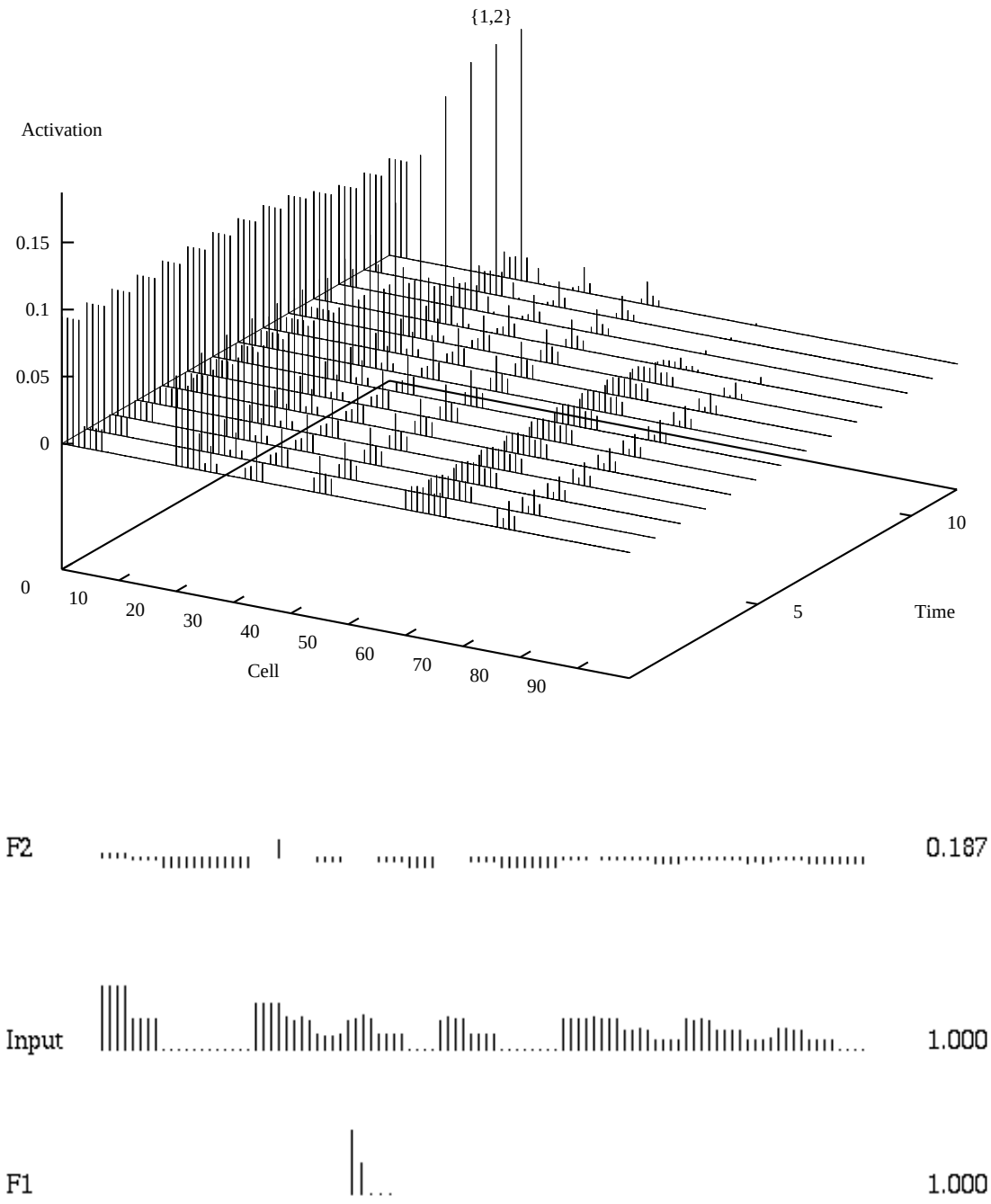


Figure 4.5: The masking field's response to the sequence $(1, 2)$ for $C = 0.125$, $F = 8704$, $f_0 = 0.16$, $g_0 = 1$.

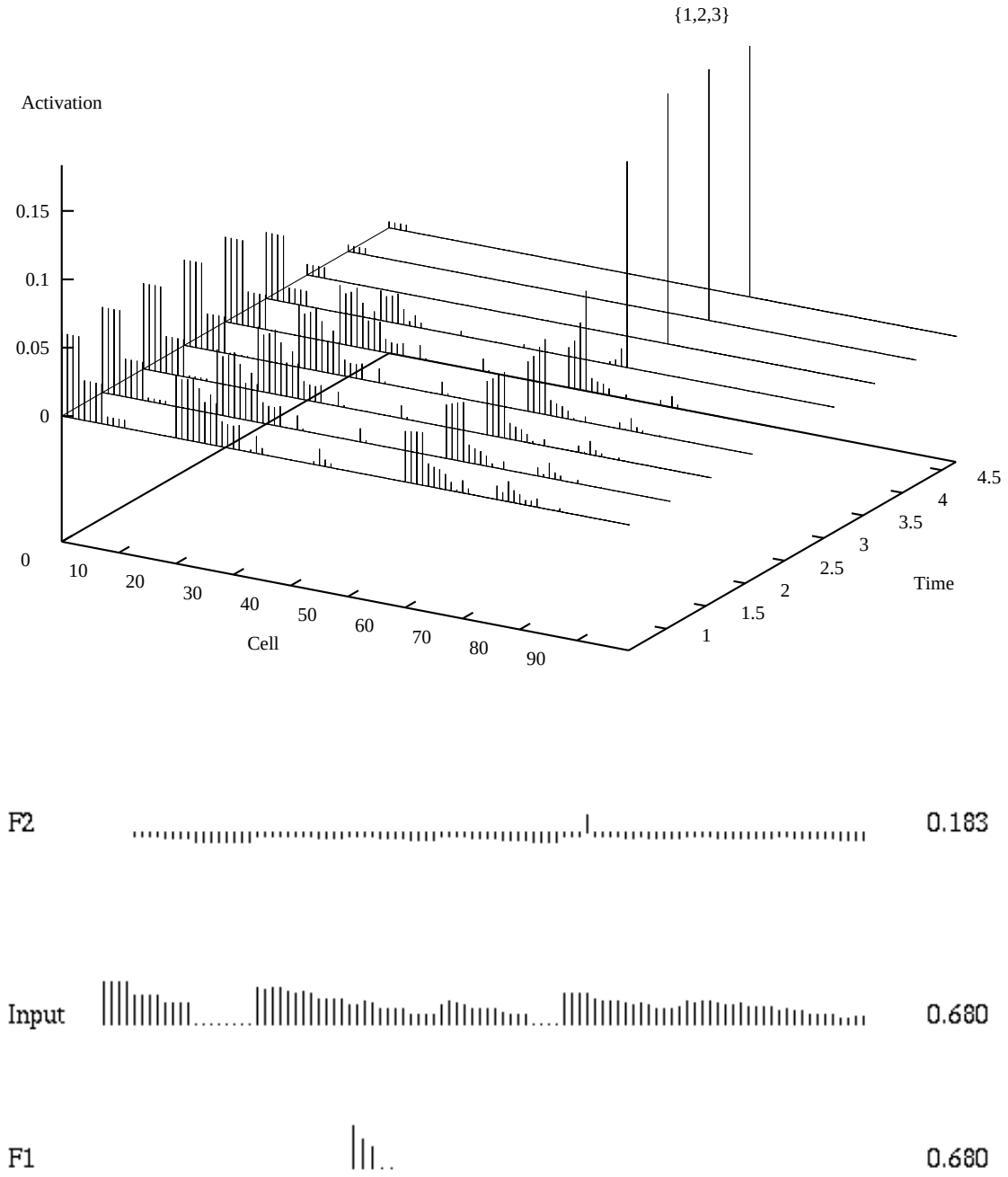


Figure 4.6: The masking field's response to the sequence $(1, 2, 3)$ for $C = 0.125$, $F = 8704$, $f_0 = 0.16$, $g_0 = 1$.

this distributed representation, the most active $\mathcal{F}_2$ cell is once again one of the cells that is connected only to the active input cells but there are also less active cells, which are connected to subsets and supersets of the set of currently active $\mathcal{F}_0$ cells. The activation of such cells represents alternative "parsings" of the sequence represented by the $\mathcal{F}_0$ pattern. In particular, the activation of $\mathcal{F}_2$ cells connected to supersets of the set of active $\mathcal{F}_0$ cells, may be thought of as embodying expectations of the way in which the sequence might continue e.g. in response to the pattern representing the sequence $(1,2)$, $\mathcal{F}_2$ cells connected to the $\mathcal{F}_0$ sets $\{1,2,3\}$,$\{1,2,4\}$ etc. might remain active at equilibrium, implying the expectation of a "3" or a "4" to follow.

Obviously *a priori* the masking field has no way of "knowing" the likelihood of a particular continuation and its initial assessment is determined by the random element present in its initial LTM traces. In order for the masking field to generate the most *plausible* parsings and expectations, consistent with the statistical regularities of a particular input environment, the LTM traces must be modified according to some learning rule. The differential equation embodying this learning rule is given in Equation 4.5. Cohen and Grossberg note that, for values of ϵ (learning rate) low enough to prevent instability, this learning rule satisfies the "adaptive sharpening property" i.e. learning tends to enhance the initial response of the network to a given input pattern. The issue of learning with respect to a masking field is, however, more subtle than this suggests. Cohen and Grossberg describe two cases in which learning should cause the masking field, in response to a given input pattern, to behave in a way that differs from its initial (a priori) response. Suppose that a particular spatial pattern (representing an item sequence), P_1, is repeatedly presented as input to a masking field so that the adaptive filter is tuned, via the learning rule, to allow a particular $\mathcal{F}_2$ cell to respond strongly to P_1. On presentation of a pattern which is itself unfamiliar but which includes P_1 as a sub-pattern, we might expect the masking field, in the light of the familiarity of P_1, to represent the sequence as "P_1 plus the rest". Similarly, if a large enough subset of P_1 is presented, which is itself "unfamiliar", we might expect the masking field cell representing P_1 to activate, eliciting an expectation for future items to "complete" P_1. This passage from [69] is particularly relevant

> " ...suppose that a given list is familiar to the network (e.g. a familiar word) but that none of its sub-lists has ever been individually presented to the network. In the $\mathcal{F}_3 \rightarrow \mathcal{F}_4$ [sic] adaptive filter, the pattern of LTM traces corresponding to

> the chunk of the whole list will be much better tuned than the LTM patterns corresponding to any sub-list chunk. This is because rapid masking of all sub-list chunks by the whole list has occurred on all learning trials ... Consequently on recall trials a sufficiently large sub-list of the list may activate the chunk corresponding to the whole list rather than its own sub-list chunk. This property is due to the differential amplification of the $\mathcal{F}_3 \rightarrow \mathcal{F}_4$ signals that correspond to the tuned LTM traces of the list chunk. This whole-list pattern completion effect should be weaker in situations wherein the sub-lists are also familiar lists (e.g., familiar word embedded in familiar word) due to three factors working together: the stronger relative amplification of the sub-list chunks by their tuned LTM patterns, the greater innate ease with which a sub-list can activate a sub-list chunk than a list chunk, and the possibility that cells with smaller parameters can mask cells with larger parameters if they receive larger inputs. These properties also indicate how a familiar word in a non-word may be recognized".

The first sentence of this extract contains an interesting paradox: it is surely not possible for a given list to be familiar to the network whilst none of its sub-lists has ever been individually presented to the network. For instance, suppose the sequence $(1, 2, 3, 4)$ is presented frequently to a network. It is obvious that the sequence $(1, 2, 3)$ must also be presented at least as often, measured in terms of frequency of occurrence. Thus the word "individually" in the above extract appears to be a statement concerning the *timing* of the input sequences. In other words, the whole list $(1, 2, 3, 4)$ is more familiar than, say, the list $(1, 2, 3)$, not because it occurs more *frequently* but because it resides in working memory ($\mathcal{F}_0$) for relatively longer. This will be the case if the inter-onset-interval (IOI) between events 3 and 4 is significantly shorter than the IOI between event 4 and the subsequent event. It will also be the case if the sequence $(1, 2, 3)$ has not been presented in alternative contexts, e.g. $(1, 2, 3, 5)$.

Applying this line of thought to the parsing of musical sequences, e.g. melodies, we might surmise that the lists or chunks which will be preferentially encoded in LTM will be those which end in notes of relatively long duration (so-called "resting notes"). Analysis of western melodies reveals that these resting notes are likely to be the notes of the tonic triad. This LTM preference for chunks ending in tonic-triad notes should lead to pattern completion effects that are directed towards those notes i.e. the tonic triad notes should be expected as

chunk or phrase endings. Furthermore, once the expected phrase ending notes are sounded, the input pattern will be in some sense "complete", there being no further expectation (at the note level), leading to what music theorists refer to as a sense of "closure".

The fact that expectation for a (musical) event facilitates the processing of that event by priming its occurrence implies that information relating to the expected event is in some way projected back down from $\mathcal{F}_2$ to $\mathcal{F}_0$. This is reminiscent of the operation of an adaptive resonance loop, in which top-down feedback is seen as a vital part of the process of stable classification. The remainder of this chapter will outline how I sought to model melodic sequence parsing and melodic expectation using a masking field embedded in an adaptive resonance architecture. The next section outlines the basics of the **ART3** algorithm and explains why it is appropriate for use in this context.

4.7 The ART3 algorithm

In the previous chapter the **ART2** algorithm was introduced, which enabled the stable classification of patterns drawn from an arbitrary input environment. This behaviour was achieved by allowing significant LTM learning to take place only at times when a sustained "STM resonance" was established around a loop consisting of two competitive-cooperative fields and the unidirectional adaptive filters linking each field to the other. The "upper" field, $\mathcal{F}_2$, used winner-takes-all dynamics to ensure that for each input pattern a "choice" of category was made by the enduring activation of that $\mathcal{F}_2$ cell whose incoming weight vector best matched the input pattern (in the sense of maximizing the inner product of the two vectors). The winning $\mathcal{F}_2$ cell then projected its "expected" pattern back down to $\mathcal{F}_1$, via the top-down ($\mathcal{F}_2 \rightarrow \mathcal{F}_1$) adaptive filter, thus initiating a resonance. This resonance was brought to an abrupt halt if the degree of mismatch between the input pattern and the expected pattern exceeded a dimensionless parameter known as the vigilance. Such a mismatch led to temporary suppression of the currently active $\mathcal{F}_2$ cell, which in turn led to a new winner of the competition for activation at $\mathcal{F}_2$. This cycle continued until an enduring resonance was established, either by the activation of an $\mathcal{F}_2$ cell at which some learning had already taken place or by the activation of a previously uncommitted $\mathcal{F}_2$ cell. In the **ART1** and **ART2** algorithms the reset of $\mathcal{F}_2$ cells is assumed to be mediated by the operation of a gated-dipole [69]. In **ART3** the reset mechanism is assumed to depend on the dynamics of neurotransmitter flow across

the synaptic cleft. There follows a brief description of this mechanism. A full description of the chemical model of the synapse and the rôle it plays in **ART3** reset can be found in Carpenter and Grossberg (1990).

The description of the synaptic model requires the definition of a number of terms

- z_{ji} is the strength of the LTM trace connecting the j^{th} $\mathcal{F}_1$ cell to the i^{th} $\mathcal{F}_2$ cell.
- u_{ji} is the amount of transmitter available for release at the corresponding synaptic cleft.
- v_{ji} is the amount of transmitter that has already been released into the synaptic cleft.
- x_i represents the activity of the i^{th} postsynaptic ($\mathcal{F}_2$) cell.

When the presynaptic transmitter is fully accumulated $u_{ji} = z_{ji}$. The amount of transmitter released into the synaptic cleft per unit time is equal to $S_j t(x_i)$, where S_j equals the signal from the j^{th} presynaptic ($\mathcal{F}_1$) cell and $t(x_i)$ is some function of the postsynaptic activity x_i. This function is constrained by the requirement that $t(0) > 0$, so that transmitter is able to reach a postsynaptic cell at resting level ($x_i = 0$). The function used by Carpenter and Grossberg is such that $t(x_i)$ is linear above a small negative threshold. Once the pre-synaptic transmitter is no longer fully accumulated, it is "topped up" at a rate proportional to the amount of transmitter depletion, so that the equation describing variation in the value of u_{ji} is

$$\frac{d}{dt}u_{ji} = (z_{ji} - u_{ji}) - u_{ji}S_j t(x_i) \tag{4.14}$$

The amount of transmitter released into the synaptic cleft, v_{ji}, is given by

$$\frac{d}{dt}v_{ji} = -v_{ji} + u_{ji}S_j t(x_i) \tag{4.15}$$

and all of this transmitter is assumed to become bound at the postsynaptic surface. The total amount of bound transmitter at the i^{th} postsynaptic cell ($\sum_{j\in J} v_{ji}$, where J is the set of $\mathcal{F}_1$ cells to which the $\mathcal{F}_2$ cell is connected), constitutes the bottom-up input to that cell. At reset the bound transmitter v_{ji} is inactivated, i.e. v_{ji} is set to zero, as is all activity in $\mathcal{F}_2$.

The processes described by the equations above mediate ART search in the following way. Assuming that initially all presynaptic transmitter is fully accumulated and all $x_i = 0$, when an input pattern is presented to $\mathcal{F}_1$, the pattern of bound transmitter at the $\mathcal{F}_2$ cells

is initially proportional to the array of dot products of the input pattern with the vectors representing the bottom-up LTM weights to each $\mathcal{F}_2$ cell. This can be seen by evaluating Equation 4.15 for $u_{ji} = z_{ji}$ and all $x_i = 0$. As before, the competitive-cooperative dynamics at $\mathcal{F}_2$ lead to the fast contrast enhancement of this initial pattern resulting (in the case of $\mathcal{F}_2$ choice) in the activation of a single $\mathcal{F}_2$ cell. In the subsequent time period, therefore, the pattern of released transmitter comes increasingly to resemble the contrast-enhanced $\mathcal{F}_2$ pattern due to the influence of the term $t(x_i)$ in the equation describing the transmitter release rate. If the activation of the winning $\mathcal{F}_2$ cell causes a mismatch reset, however, the bound transmitter is neutralized. Since the majority of this transmitter is bound to the $\mathcal{F}_2$ cell which has erroneously classified the current pattern, in the time period after reset the new pattern of released transmitter will no longer be proportional to the inner product **S.z** but will be proportional to **S.u** and as such will be selectively biased against re-exciting the $\mathcal{F}_2$ cell which was previously active and whose activation resulted in mismatch reset. Carpenter and Grossberg [23] present the results of simulations of **ART3** behaviour which demonstrate the efficiency of such a search mechanism. The **ART3** algorithm has certain advantages over **ART2**, particularly in situations where the dynamics of the $\mathcal{F}_2$ field do not necessarily lead to a single winning $\mathcal{F}_2$ cell i.e. when a distributed representation is present at $\mathcal{F}_2$. Such a distributed representation is characteristic of a masking field with particular parameters and it was decided to use an **ART3** type algorithm when embedding the masking field in an adaptive resonance architecture. For reasons similar to those discussed in relation to the masking field above, I wrote a program which reproduced the published **ART3** results of Carpenter and Grossberg before adapting the algorithm to my particular application.

There is a slight difference between my implementation of **ART3** and that described in [23], namely that in that paper they employ the transmitter equations described above to model the synaptic behaviour in both the bottom-up *and top-down* adaptive filters. Since these equations are primarily directed towards the establishment of a mechanism for post-reset search it is unnecessary to extend their use to the synapses of the top-down adaptive filter; as Gjerdingen has pointed out (personal communication and [65]), the employment of the **ART3** synaptic model in the $\mathcal{F}_2 \rightarrow \mathcal{F}_1$ adaptive filter can lead to instability in regimes in which the vigilance parameter is set high. A further consideration relates to the computational overhead required to perform the integration of the transmitter equations—this is justified in

the case of the bottom-up adaptive filter, where, as has been demonstrated, desirable effects can accrue, but is not justified in the case of the top-down adaptive filter. For these reasons the operation of the $\mathcal{F}_2 \rightarrow \mathcal{F}_1$ adaptive filter in my program will be the same as that in the **ART2** algorithm.

The next section will show how the basic masking field model must be extended somewhat to allow it to process basic melodic patterns.

4.8 Extending the masking field

As mentioned above, the masking field used in the initial simulations was not sufficient for modelling the perception of even simple melodies. It was decided that a group of twelve well-known nursery rhymes would constitute an appropriate set of examples of simple western tonal melodies, from which it was hoped that the network would be able to learn typical melodic progressions and the expectations that they imply. The pitch data relating to the nursery rhymes was coded in terms of one octave of scale-notes i.e. do, re, mi, fa, so, la, ti, and was thus in a transposition-invariant format. The data file describing the melodies also contained duration information measured relative to the duration of a single "beat". More details regarding the input data will be presented in the next chapter. This section, however, will deal only with the constraints which this choice of input data places on the size and structure of the masking field. Obviously, it was necessary to increase the number of cells in the input field from 5 to 7, each cell representing one of the scale-notes. In addition, it was decided to increase the number of sections in $\mathcal{F}_2$ from 3 to 4. There were a number of reasons for this. Firstly, it has been widely observed that item sequences, including musical phrases, are best recalled when perceptually segmented by the Subject into chunks of three or four items each ([56], [82], [41]). Secondly, as mentioned earlier, Grossberg has related the maximal (optimal) chunk size to the Transient Memory Span of STM, which he assumes to be approx equal to 4. It thus seemed necessary to allow the masking field to represent chunks of up to four items in length. Indeed, preliminary results presented in conjunction with [120] suggested that such a modification would increase the usefulness of the network. A further modification to Cohen and Grossberg's original masking field formulation was to ensure that the operation of the working memory was consistent with the LTM Invariance Principle and the Normalization Rule described earlier in the chapter. For example, in Cohen

and Grossberg's paper the activation patterns used at the working memory were $(1.5, 0, 0, 0, 0)$, $(1, 0.5, 0, 0, 0)$ and $(0.68, 0.48, 0.34, 0, 0)$, which, although "sum-normalized", are not consistent with the constraint that the introduction of a new event into the working memory should only alter the existing activations by a single multiplicative (shunting) value. In the network developed here (known rather unimaginatively as **MASKART**) the vector representing the working memory activities was length-normalized and these activities were multiplied by a constant shunting parameter, w, at the onset of a new event.

Many trials were conducted in an attempt to ascertain a set of parameters for which the extended masking field would behave in a manner consistent with the sequence masking principle. There are seven parameters in Cohen and Grossberg's original simulations which affect the dynamics of the masking field, viz. A, B, C, D, F, f_0 and g_0. Cohen and Grossberg give little rationale for the setting of these parameters so, as a strategy for searching the "parameter space", I chose to reduce the degrees of freedom by fixing some of the parameters at those values used by Cohen and Grossberg. Thus in all the simulations $A = 1$, $B = 1$, $D = 4$, $f_0 = 0.16$, $g_0 = 1$. In addition, for the "choice" condition, C was set to 1, allowing the behaviour of the masking field to be investigated by controlling the single parameter F. It was discovered that Cohen and Grossberg had set F to be twice the maximum total inhibitory weight to any $\mathcal{F}_2$ cell, so the equivalent value for the extended masking field was used as an initial value of F. A large number of simulations were run with the shunting parameter, w, set to 1.5, but satisfactory results were not obtained. Analysis of the simulations showed that this was due to the comparatively small effect that the activation of the $\mathcal{F}_0$ cell corresponding to the last list item had on the pattern of input to $\mathcal{F}_2$. Nigrin makes a similar point in Chapter 4 of his thesis [117]. Satisfactory results were obtained, however, by reducing w to 1.2 and F to 3500 which gave the results shown in Figures 4.7-4.10. As before, a more distributed pattern of equilibrium activation could be obtained by altering the parameters C and F—in this case the settings $C = 0.2$ and $F = 24000$ were used. The reduction of w to a level approaching unity has a disadvantage, however, namely that, under these conditions, the vigilance parameter has to be set extremely high if one wishes the network to be able to distinguish between patterns consisting of the same items in different temporal order. The trade-off between allowing the last item in a list to have a significant effect on the input pattern at $\mathcal{F}_2$, which suggests a value of w approaching unity, and the need to be able to

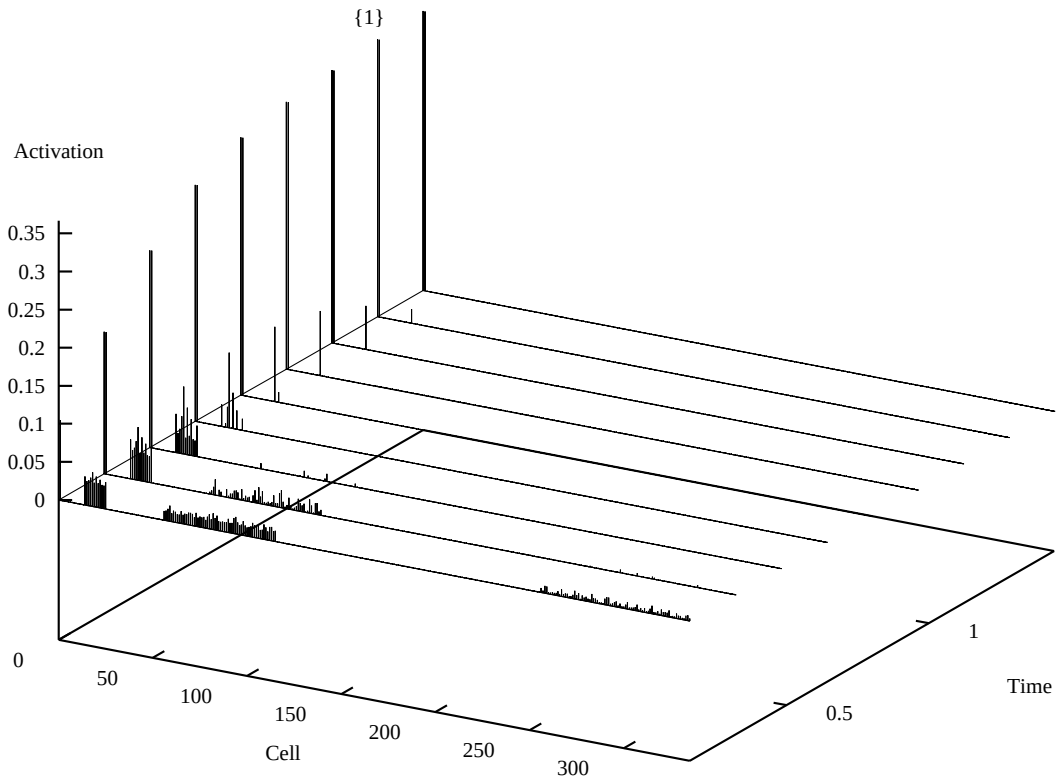


Figure 4.7: The extended masking field's response to the pattern (1) (see text for parameters)

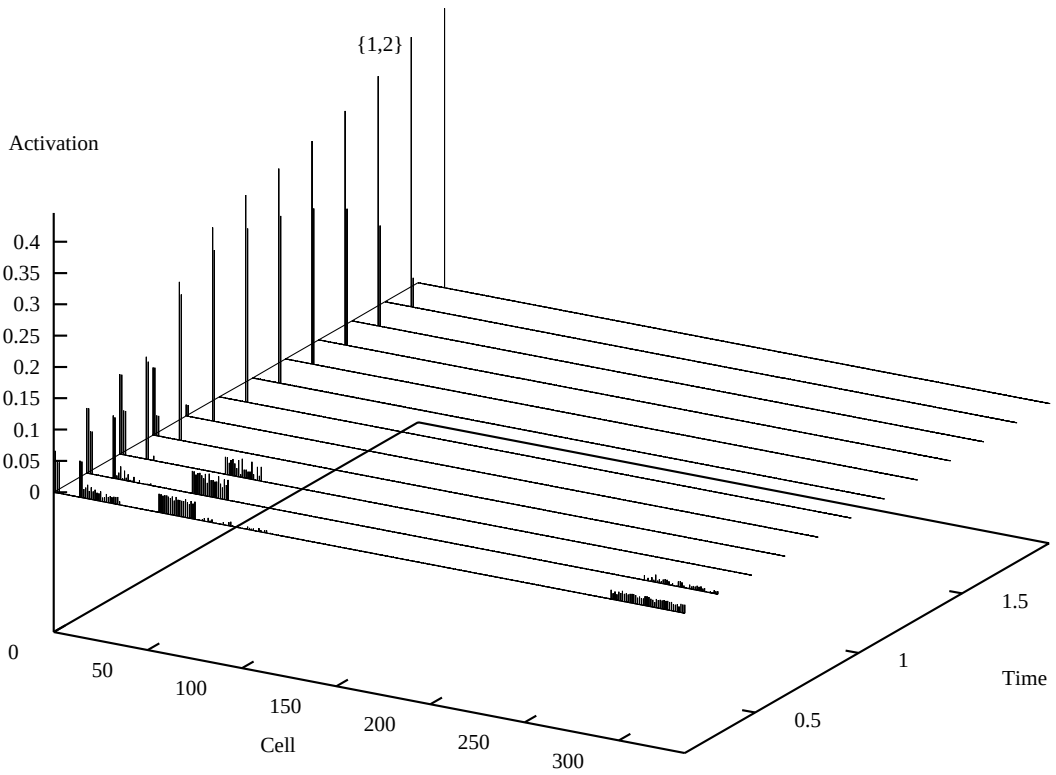


Figure 4.8: The extended masking field's response to the pattern $(1, 2)$ (see text for parameters)

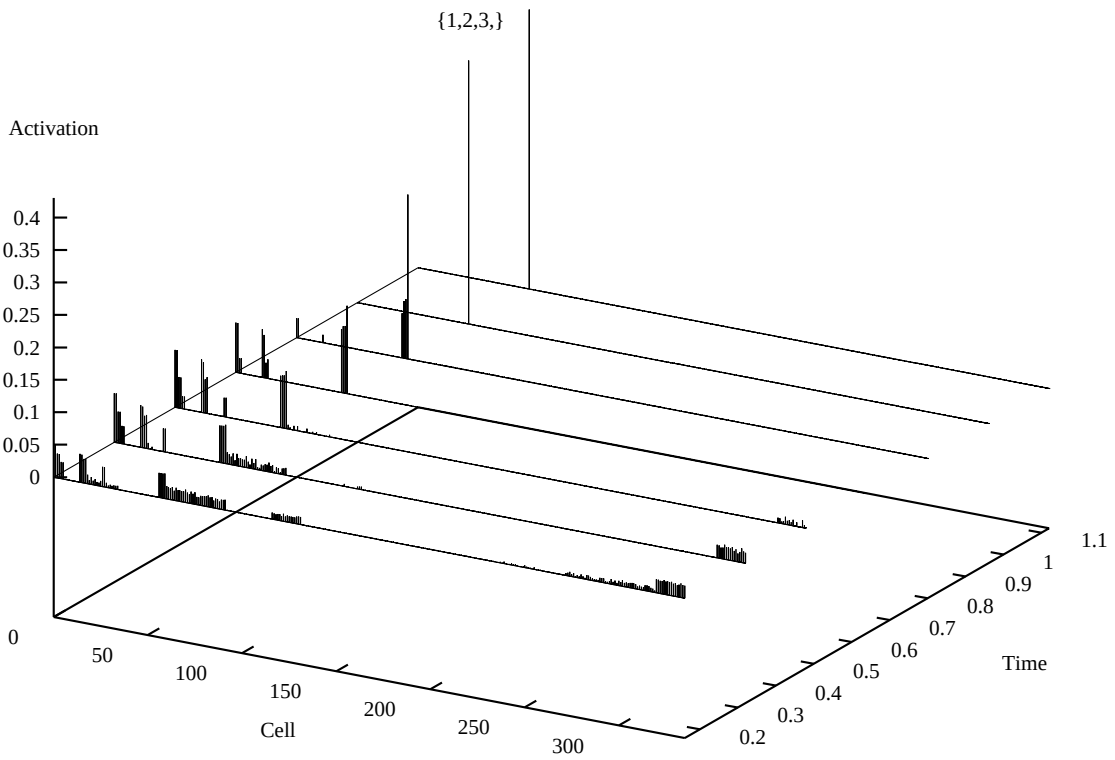


Figure 4.9: The extended masking field's response to the pattern $(1, 2, 3)$ (see text for parameters)

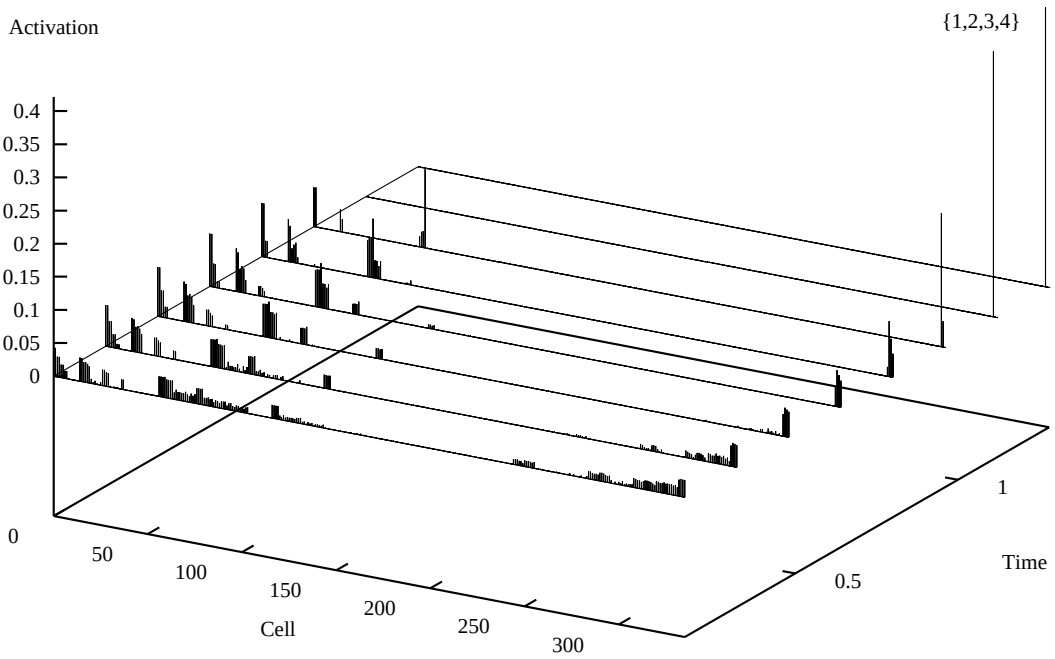


Figure 4.10: The extended masking field's response to the pattern $(1, 2, 3, 4)$ (see text for parameters)

distinguish clearly between different orderings of the same items, which suggests a relatively high w, involves delicate balancing of parameters. I believe that this lack of robustness in the setting of **MASKART**'s parameters has profound implications for the wider applicability of such networks. A later chapter will introduce the work of Nigrin [117], which attempts to deal with these problems by introducing additional complexity into the process of feeding activations forward via an adaptive filter.

Finally, in order to embed the masking field in an adaptive resonance loop, it is necessary to employ a top-down adaptive filter linking the masking field to the working memory. The provision of this top-down adaptive filter allows learning of patterns in a manner that solves the stability-plasticity dilemma mentioned in the previous chapter. In particular, the ability of top-down expectations to influence the course of learning is clearly necessary to allow the network to learn to distinguish patterns corresponding to sequences containing the same items in different orders. For instance, suppose that the network is repeatedly exposed to the pattern representing the sequence $(1, 2, 3, 4)$. Learning in bottom-up LTM traces will allow one $\mathcal{F}_2$ cell to respond preferentially to this pattern by strengthening the weights of connections between that cell and the relevant $\mathcal{F}_0$ cells. Now suppose that the pattern representing the sequence $(4, 3, 2, 1)$ is subsequently active in working memory for the first time. As a result of the learning described above it is very likely that the same $\mathcal{F}_2$ cell will initially respond to this sequence. If the activation of this $\mathcal{F}_2$ cell is allowed to persist, its LTM weights will begin to become parallel to the second $\mathcal{F}_0$ pattern, thus overriding its previous learning. This behaviour is obviously unstable and may lead to a single $\mathcal{F}_2$ cell responding to a variety of distinct $\mathcal{F}_1$ patterns. As discussed in the previous chapter, an adaptive resonance architecture allows such a situation to be avoided by allowing the $\mathcal{F}_2$ cell to be reset before any significant learning can be achieved, if the current $\mathcal{F}_0$ pattern does not match the cell's expectations to a degree that satisfies the vigilance criterion. As noted previously, the vigilance parameter must be set to a value very close to one if the network is to distinguish between patterns representing different orderings of the same items, particularly when the shunting parameter approaches unity. The differential equation that describes the learning of top-down weights is similar to that describing bottom-up weight learning

$$\frac{d}{dt} z_{ij} = \epsilon f(x_i)(LI_j - z_{ij}) \tag{4.16}$$

the main difference being that the top-down weights are initially equal to zero.

Figure 4.11 shows how the working memory field and the masking field are integrated within an **ART3** architecture. An intervening field, $\mathcal{F}_1$, made up of three levels ($1A$, $1B$ and $1C$), allows for the merging of bottom-up and top-down information, whilst buffering the working memory against direct top-down influence. The degree of match between the resulting, merged pattern, active at level $1B$, and the working memory pattern, can be compared with the vigilance parameter to ensure the stable learning which is characteristic of ART networks.

The equations describing $\mathcal{F}_1$ activities are as follows

$$\mathbf{x}_{1C} = \mathbf{I} + p_1.t(\mathbf{y}_{1B}) \tag{4.17}$$

$$\mathbf{x}_{1B} = t(\mathbf{y}_{1C}) + p_2.t(\mathbf{y}_{1A}) \tag{4.18}$$

$$\mathbf{x}_{1A} = t(\mathbf{y}_{1B}) + p_4.\mathbf{d} \tag{4.19}$$

where the function t quenches any activities below the intra-field threshold and the vector $\mathbf{d}$ represents the top-down input from the masking field. The activities $\mathbf{y}_{1A}$, $\mathbf{y}_{1B}$ and $\mathbf{y}_{1C}$ are simply the length-normalized versions of the corresponding $\mathbf{x}$ activities.

4.9 Constraints on the operation of a masking field within ART3

4.9.1 "Input" and "Mismatch" Reset

It will be seen that many of the constraints put on the operation of the masking field within an **ART3** network concern the conditions under which reset should occur. Reset involves the neutralization of postsynaptic bound transmitter and the setting of all postsynaptic activations to zero—in general terms this should occur when the active masking field cell does not properly represent the current pattern in working memory, relative to the vigilance criterion. Carpenter and Grossberg [23] identify two types of reset, each of which relies on a slightly different cause for such a misrepresentation. The first is called a "mismatch reset", which occurs when the $\mathcal{F}_2$ cell that responds to a given input pattern is inappropriate for describing that input pattern—this is the familiar ART reset. The second type is called an "input reset" which occurs when

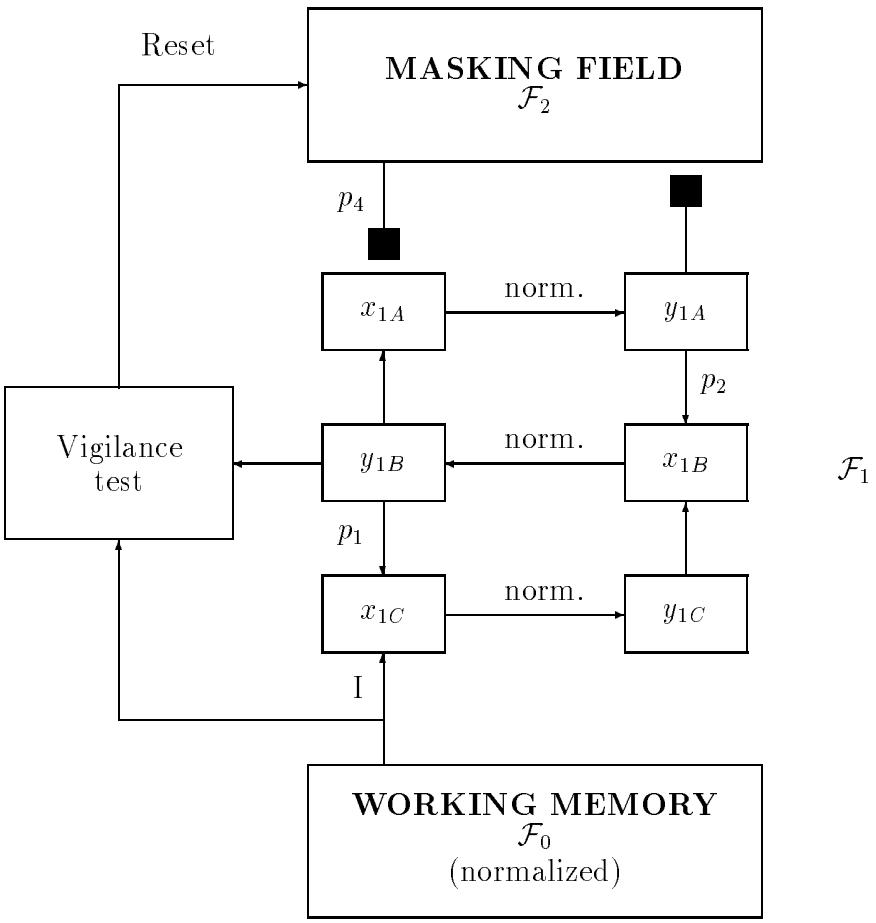


Figure 4.11: The **MASKART** network (see text for details)

the input pattern changes so that an $\mathcal{F}_2$ cell that was formerly appropriate suddenly becomes inappropriate. The distinction between the two types of reset is a subtle one since they are both precipitated by a given degree of mismatch between the bottom-up (working memory) pattern and the top-down (expectation) pattern. After initial simulations, however, in which slow learning of sequences was attempted, it became clear that some clarification of the mechanisms for input and mismatch reset was necessary. To illustrate this point it will be necessary to examine two different situations with which the network might be presented. It should be noted that, although these situations are presented here as "thought experiments" to illustrate constraints under which the network must operate, they are situations which arose, and led to problems, in actual simulations.

Hereafter, the term "the pattern $(1,2,3,4)$" will refer to the spatial pattern present in working memory after presentation of the sequence of items $(1,2,3,4)$. An $\mathcal{F}_2$ cell will be said to "represent" this pattern if LTM learning has led to that cell activating strongly in response to the pattern. In addition, the phrase "the activation of an item in working memory" will be used as a shorthand for the activation of the working memory cell that responds to that item.

Suppose that at some point in time the pattern $(1,2,3,4)$ has been well learned due to frequent "isolated" repetition. Now assume that this learning is such that on presentation of the sequence $(1,2,3)$, the cell representing the complete pattern $(1,2,3,4)$ activates, eliciting an expectation for the item (4). This expectation will be indicated by a state of affairs in which the pattern $(1,2,3,4)$ will be active, or partially active, at the top level of $\mathcal{F}_1$ (level $1A$), whilst the pattern $(1,2,3)$ will be active at the central level of $\mathcal{F}_1$ (level $1B$). This situation can only come about by strict application of the "$\frac{2}{3}$ rule" [70] to cells at level $1B$. The $\frac{2}{3}$ rule says that for a cell to be activated at this level whilst a top-down expectation pattern is active, it must receive both top-down and bottom-up input. The version of the $\frac{2}{3}$ rule employed in the **ART2** and **ART3** networks is a weak $\frac{2}{3}$ rule: the high level of intra-field feedback guarantees that the activation of any cell which receives bottom-up input but no top-down input will be quenched. However, this can lead to situations in which a cell activates due to top-down input alone. Carpenter and Grossberg [22] admit this possibility. In the case of **MASKART** the weak $\frac{2}{3}$ rule is not sufficient since the activation at level $1B$ of an item which is expected to occur but has not yet done so, could lead to inappropriate

mismatch reset. Thus we have our first constraint, namely that the $\frac{2}{3}$ rule is strictly enforced at level $1B$. This can be easily achieved by gating the effect of top-down feedback on cells at this level by the activation of the cells themselves. The ramification of this is that the appropriateness of a representation at $\mathcal{F}_2$ is only judged, relative to the vigilance criterion, over the set of items that have already occurred, and that the validity of any "predictions" that are made is only assessed as subsequent events unfold. The fact that expectation can be expressed as the difference between the pattern at the level $1A$, which is influenced by the expectation pattern of the active $\mathcal{F}_2$ cell, and the pattern at the middle level of $\mathcal{F}_1$, is useful since we can clearly distinguish between items which are expected to happen and items which have already happened.

Returning to our example, in which the sequence $(1, 2, 3)$ has elicited expectation of the item (4), it can now be seen that if the expected item occurs and activates in working memory, $\mathcal{F}_0$, it is extremely undesirable for an input reset to occur, in spite of the fact that there is a momentary mismatch between the pattern at working memory and that at level $1B$. Such a transmitter reset would bias the network against reactivating the $\mathcal{F}_2$ cell representing the pattern $(1, 2, 3, 4)$, even though this is obviously the correct cell to activate, since it represents the working memory pattern exactly. Thus we have our second constraint: that the onset of a new item in working memory should not cause immediate transmitter reset. In order that this should be the case, the new working memory pattern must be able to establish itself at level $1B$ so as to eliminate the mismatch between the two patterns, and while this is occurring mismatch reset should be inhibited. This can be quickly achieved by eliminating the feedback between level $1B$ and level $1C$, either by resetting the activity of all $\mathcal{F}_1$ cells when a new item enters working memory or, alternatively, by resetting only some of them. Analysis of a second situation, with which the network is likely to be presented, will illustrate my reasons for choosing the latter option.

Having ascertained that the advent of a new item in working memory should not precipitate immediate transmitter reset, it can be seen that transmitter reset will only occur, after the new pattern has been established at level $1B$, if the top-down influence is sufficient, under the $\frac{2}{3}$ rule, to distort the pattern at that level sufficiently to cause mismatch. If, on the onset of a new item, all activations at $\mathcal{F}_1$ were to be set to zero before establishing the new working memory pattern, then such a mismatch could only occur if the currently active $\mathcal{F}_2$ cell projects

a *significant* inappropriate expectation pattern to $\mathcal{F}_1$. In the initial stages of a slow learning regime, however, this is unlikely to be the case, since all top-down weights are initially equal to zero. Under slow learning conditions it may be necessary to wait many trials before sufficient top-down traces exist to cause transmitter reset. For instance, suppose that the sequence $(1, 2, 3, 4)$ is to be presented to the network; after the (1) has been presented, a size-1 cell should activate in $\mathcal{F}_2$ (assuming no prior learning). Due to the constraint imposed above, the arrival of the item (2) in working memory, should not be able to cause transmitter reset by itself, in spite of the fact that it introduces a momentary mismatch between the pattern in working memory and the normalized pattern in level $1B$. The new working memory pattern is able, due to the resetting of $\mathcal{F}_1$ intra-field feedback, to establish itself at level $1B$. This will lead to a situation in which the size-1 $\mathcal{F}_2$ cell will continue to be active at $\mathcal{F}_2$ since it does not yet generate an expectation large enough to cause mismatch. Gjerdingen (personal communication) has pointed out that this behaviour will eventually cease when sufficient top-down learning has taken place at this cell. Nevertheless the fact that the $\mathcal{F}_2$ cell that responds to the first item of a given sequence gains a considerable head start, in terms of its LTM learning, appears to be problematic. I propose that this can be avoided by only resetting a particular set of $\mathcal{F}_1$ cells on item onset. The minimum requirement is that the feedback from the central level of $\mathcal{F}_1$ to the bottom layer is set to zero to allow the new working memory pattern to establish itself at level $1B$, therefore I propose that on item onset all activation at level $1B$ is quenched to zero. This has a two-fold effect: firstly it inhibits mismatch reset as required, since the degree of match is always unity when one of the vectors is zero; secondly it allows the new working memory pattern to attempt to establish itself at level $1B$. The success with which this can occur depends on the pattern that persists at the upper level of $\mathcal{F}_1$. If this pattern is influenced by top-down activation from $\mathcal{F}_2$ then the pattern in working memory will only be enduringly established at level $1B$ if the active $\mathcal{F}_2$ cell is appropriate i.e. its expectation pattern is fully consistent with all the items in working memory—in this case there will be no transmitter reset when level $1B$ is reactivated. If the $\mathcal{F}_2$ cell projects an inappropriate expectation pattern then the working memory pattern will not be established at level $1B$ leading to mismatch reset. Likewise, if there is no significant top-down expectation from $\mathcal{F}_2$ then the *previous* working memory pattern will persist at the upper level of $\mathcal{F}_1$ and will dominate subsequent activation at level $1B$, once again leading to mismatch reset.

Thus the simple requirement that activation at level $1B$ be set to zero at the onset of a new item, combined with the application of a strict $\frac{2}{3}$ rule at that level, allows the network to cope with both the potentially problematic situations described above and clarifies the processes underlying both input and mismatch reset.

4.9.2 ART3 dynamics

A further problem arises with the **ART3** transmitter dynamics described in [23]. In that paper an $\mathcal{F}_2$ cell which has caused mismatch reset is able to reactivate at $\mathcal{F}_2$ several times before the depletion of its transmitter is sufficient to prevent it from continuing to activate. The process of repeated resets is time consuming but not, in itself, disastrous. However, once the $\mathcal{F}_2$ cell has had the levels of its incoming transmitter depleted to the extent that it no longer wins the competition for activation at $\mathcal{F}_2$, it can immediately begin to replenish its pool of presynaptic transmitter via the term $(z_{ji} - u_{ji})$ in Equation 4.14. This can lead to a problem if there are several cells which erroneously activate in succession in $\mathcal{F}_2$—once the latter cells have been through a series of resets it is quite possible that the presynaptic transmitter at the original erroneous cell has accumulated enough to allow this cell to activate once more. In order to remove this possibility, I altered the transmitter dynamics slightly so that the process of transmitter re-accumulation was temporarily inhibited at the synapses abutting an $\mathcal{F}_2$ cell that had already erroneously activated for the current input pattern; for convenience, this inhibition was assumed to endure until the onset of the next item. This alteration was not sufficient to prevent the undesirable repeated activation described above, so a further modification was made: if activation of an $\mathcal{F}_2$ cell caused a mismatch reset during presentation of a given pattern then, for the time before the next item onset, all outgoing signals from that cell would be set to zero. In particular, the cell is unable to emit inhibitory signals which might otherwise prevent alternative $\mathcal{F}_2$ cells from activating. This effect can be thought of as transmitter depletion at the cell's efferent inhibitory synapses, such depletion being greatest at those cells with which the cell had formerly competed most strongly.

4.9.3 Learning should be slowed when expectations are active

Another constraint which was suggested by initial simulations was once again concerned with the operation of the network after sufficient learning has taken place to elicit expectations.

Again assume that the pattern $(1,2,3,4)$ is just sufficiently familiar that on presentation of the sequence $(1,2,3)$ the $\mathcal{F}_2$ cell corresponding to the pattern $(1,2,3,4)$ activates. In this situation it is possible that the pattern of activation at level $1A$ is not exactly parallel to the pattern that will be present in working memory after the last item, 4, has occurred. This is because the pattern in level $1A$ is an "amalgam" of the normalized pattern at level $1B$, $(1,2,3)$, and the top-down expectation pattern, $(1,2,3,4)$. As learning proceeds the top-down expectation will come to dominate this amalgam but in the early to mid stages of learning such dominance cannot be guaranteed. In this case there is a possible problem: when an $\mathcal{F}_2$ cell is active its LTM traces will tend to become parallel to the pattern simultaneously active at the top level of $\mathcal{F}_1$ (see Equations 4.5 and 4.16). Thus if learning were allowed to proceed as usual whilst an $\mathcal{F}_2$ cell representing a superset of the current working memory pattern was active, then such learning may lead to instability. To avoid this possibility, the learning rate is reduced by a large factor (e.g. 100) at times when a sustained expectation is active.

4.9.4 The equilibrium size of bottom-up weight vectors

According to equation 4.5, the length of the bottom-up weight vector to a given $\mathcal{F}_2$ cell tends, during LTM learning, towards a multiple, L, of the length of the input vector which it comes to represent. In **MASKART** the input vector is length-normalized (its length = 1), so the length of the bottom-up weight vector tends towards L, which Cohen and Grossberg set equal to 10. This leads to the following problem: suppose that the sequences (1) and $(1,2,3)$ are presented to the network with sufficient frequency that each becomes represented by an $\mathcal{F}_2$ cell. As LTM learning proceeds the single bottom-up weight to the size-1 $\mathcal{F}_2$ cell will tend towards a value of $z = 10$, whilst the bottom-up weight vector to the size-3 $\mathcal{F}_2$ cell will tend towards $\mathbf{z} = (6.78, 5.65, 4.71)$, for $w = 1.2$. When these equilibrium values are reached, however, the inputs to the two cells on presentation of the pattern $(1,2,3)$ will be as follows: $I_{(1)} = 6.78$ and $I_{(1,2,3)} = \frac{10}{3} = 3.33$, where I_P denotes the input to the $\mathcal{F}_2$ cell that represents the pattern P. The denominator in the fraction $\frac{10}{3}$ is due to the dilution of the input to the size-3 cell embodied in the term p_{ji} in Equations 4.4 and 4.11. Thus, on presentation of the input pattern $(1,2,3)$, the input to the size-1 cell is more than twice as much as the input to the size-3 cell, even though the size-3 cell has "fully learned" the input pattern. The competitive advantage enjoyed by the larger $\mathcal{F}_2$ cell is not sufficient to compensate for this

disparity in inputs and the size-1 cell activates at $\mathcal{F}_2$. Compare this with the initial situation, in which all bottom-up LTM weights are equal: the equivalent ratio $I_{(1,2,3)} : I_{(1)}$ is equal to 0.84, an input ratio which is sufficiently high to allow the larger $\mathcal{F}_2$ cell to activate.

To avoid this problem, I propose that the value of L in Equation 4.5 should vary with the size of the relevant $\mathcal{F}_2$ cell. Initial simulations used a linear function: the value of L_j was set to the size of the j^{th} $\mathcal{F}_2$ cell. This, however, was found to overcompensate for the problem described above, leading to a situation in which a larger $\mathcal{F}_2$ cell that had fully learned an input pattern, could always mask a smaller $\mathcal{F}_2$ cell, that had *previously* fully learned a subset of that pattern, when the smaller pattern was presented. A second, slower-than-linear function, for which L_j was set to a constant plus the square root of the size of the relevant $\mathcal{F}_2$ cell, proved satisfactory and was the function used in all simulations. Equation 4.5 should therefore be replaced by

$$\frac{d}{dt} z_{ji}^{(J)} = \epsilon f(x_i^{(J)})(L_J I_j - z_{ji}^{(J)}) \tag{4.20}$$

$$L_J = 3 + \sqrt{|J|} \tag{4.21}$$

4.9.5 Resets

Simulations also suggested constraints on the behaviour of **MASKART** after transmitter reset had occurred. In particular, immediately after transmitter reset, a new competition for activation takes place at $\mathcal{F}_2$. In some cases two cells will compete strongly and there will be a short time, before one of the cells dominates, during which both of them have significant activation. If, as is likely, each projects a significant top-down expectation to $\mathcal{F}_1$, the *sum* of these expectations might distort the activity at level $1B$ to the extent that a further mismatch reset occurs a short time after the first. Such behaviour is undesirable, since it could lead to a bias against the activation of an appropriate $\mathcal{F}_2$ cell. To prevent this from occurring, the reset signal was disabled for a fixed time period after firing to allow the activation at $\mathcal{F}_2$ to stabilize before the adequacy of the newly activated $\mathcal{F}_2$ cell, measured as the match between its expectation pattern and the current working memory pattern, could be assessed. Thus the cell that generates the reset signal can be thought of as having a "post-firing" refractory period.

A second, related problem concerns the ability of the network to learn a pattern similar to one previously learned. Suppose the pattern $(1, 2, 3)$ has been learned by the network. On subsequent presentation of a similar, but novel, pattern, say $(1, 3, 2)$, the $\mathcal{F}_2$ cell representing the more familiar pattern is likely to activate erroneously. After the reset refractory period (that follows the "input reset" that occurs on presentation of the third item) is over, a further reset will therefore occur and and a new competition for activation will proceed. It is possible that a further mismatch reset will occur before an appropriate, uncommitted $\mathcal{F}_2$ cell is able to activate, by which time there may only be a short period, during which this cell can begin to learn the current pattern, before the next input item causes a further "input reset". Thus it will take longer (in terms of the number of presentations) to learn a pattern similar to one already learned.

It is possible to prevent this problem by ensuring that the effective presentation time of a pattern is extended each time a reset occurs. The extension of the presentation time is limited by the onset of the next item. Such a modification is easily accomplished within the **MASKART** program, but is less easily accounted for in terms of the behaviour of a real network. One has to assume that the network is only permitted to change its state during a time "window" after item onset. Nigrin [117] suggests a similar mechanism for making his network "event driven" rather than "time driven". The magnitude of this time window is necessarily less than the inter-onset-interval between successive items, and is able to be increased by a reset event. The operation of this type of mechanism is discussed in more detail in Chapter 6.

4.10 Constraints on the operation of $\mathcal{F}_0$

4.10.1 Limiting the capacity of working memory

In the early part of this chapter a mechanism was described for allowing sequence information to be represented by a spatial pattern of activity, in which temporal order was represented by relative cell activation. This mechanism was used by Grossberg [68] to explain various data derived from recall experiments. These data indicate the appearance of a bow in serial recall, which was modelled using to the LTM Invariance Principle and the Normalization Rule. Clearly the appearance of a bow in **MASKART**'s working memory ($\mathcal{F}_0$) would distort

its learning process by disrupting item order information, so it was decided to arrange $\mathcal{F}_0$ so as to exclude this possibility. The simplest solution is adopted, namely that at the onset of a given item, the $\mathcal{F}_0$ cell corresponding to that item is assigned an activation of 1 and all other items are multiplied by the shunting parameter, w. Clearly, using this method, there appears to be no limit to the capacity of working memory. This is unrealistic, so a solution adapted from that proposed by Nigrin [117] is adopted: when the total number of events in working memory reaches a certain number (this can be surmised from the total working memory activation), an $\mathcal{F}_0$ reset event is triggered, during which any cell with an activity *above* a given threshold is reset to zero activity. By varying the threshold, therefore, a varying number of items can be made to "drop out" of working memory. Nigrin introduces a random element in the determination of this number. Details of an alternative scheme, more appropriate to the specific musical input, will be described later. It should also be noted that an $\mathcal{F}_0$ reset must necessarily trigger an $\mathcal{F}_1 \rightarrow \mathcal{F}_2$ transmitter reset.

4.10.2 The "banana problem"

A fundamental problem with this method of storing sequence information concerns the issue of repeated notes. For instance, it is impossible to represent the sequence $(1, 2, 3, 1)$ with an $\mathcal{F}_0$ spatial pattern. Marshall (personal communication) refers to this as the "banana" problem, a name which alludes to the difficulty of distinguishing the working memory representations of the utterances "banana" and "bana". Grossberg does not mention the problem in his original paper, but Nigrin [117] deals with it in some depth, though at the cost of introducing much complexity. Nigrin's proposed (theoretical) solution is implemented in Chapter 7. In the present chapter and the following two chapters a more simple approach is taken: if an item occurs whose corresponding $\mathcal{F}_0$ cell is already active, all $\mathcal{F}_0$ cells with activation greater than that cell are reset and the activation of the cell itself is set to 1. In this way the item order information remains correctly represented in $\mathcal{F}_0$. There is one exception to this rule: an immediate repetition of a note is treated as an extension of that note i.e. if a given note sounds twice consecutively then it is treated as a single note of duration equal to the sum of the two original note durations. Once again this rule is necessary to allow correct order information to be retained in working memory under constraints imposed by the "banana" problem. The $\mathcal{F}_0$ reset necessitated by the non-immediate repetition of an item also triggers

transmitter reset.

4.11 The setting of parameters

For completeness, in this section I shall briefly detail some of the values used in the **MASKART** simulations and, more importantly, the reasons for these choices—there are many parameters involved in the definition of ART networks and masking fields and it is therefore important to derive some heuristics for their setting.

For the **ART3** part of **MASKART** the important parameters are the vigilance, inter-field and intra-field feedback strengths, the intra-field quenching threshold, the transmitter release function and the initial bottom-up LTM traces, z_{ji}.

In order that **MASKART** can learn to distinguish between different orderings of the same items, the vigilance parameter must be set very high indeed. For instance, if we assume that the shunting parameter, w, is equal to 1.2 (as it was in all simulations) then, if we require there to be a mismatch reset when the working memory pattern $(1,2,3,4)$ is erroneously classified by the $\mathcal{F}_2$ cell which has fully learned to respond to the sequence $(1,2,4,3)$, the vigilance must be set to a value of 0.999. Similarly, to distinguish $(1,2,3)$ from $(1,3,2)$ requires a vigilance of 0.998 and to distinguish $(1,2)$ from $(2,1)$ a vigilance of 0.996. As noted earlier, these very high values of vigilance are required as a result of the necessity to choose w close to 1, a constraint which was itself imposed by the need to ensure the correct behaviour of the masking field under various conditions. In the simulations presented a vigilance of 0.999 was used.

The intra-field feedback parameters p_1 and p_2 (see Figure 4.11) were set so that top-down signals were approximately an order of magnitude larger than bottom-up signals. This choice of parameters satisfies the **ART2** stability constraint that refers to the "weak $\frac{2}{3}$ rule". The inter-field feedback parameter, p_4, is set in a similar way. In the following simulations $p_1 = p_2 = 10$ and $p_4 = 5$.

The inter-field quenching threshold is less relevant in **MASKART**, where $\mathcal{F}_0$ and $\mathcal{F}_1$ patterns represent item sequences, than in **ART2** or **ART3** where the spatial patterns at these fields can represent other types of information. In short, the quenching threshold must be set low enough so that no item information is lost during preprocessing at these levels and high enough that the $\mathcal{F}_0$ and $\mathcal{F}_1$ fields could operate in the presence of "noise". In the

simulations, the quenching threshold was set to 0.1.

The constraint on the transmitter release function, $t(x)$, is that it is equal to a small positive value for $x = 0$ and that it grows for increasing x in a manner that is at least as fast as linear. For values of x significantly less than zero $t(x)$ should equal zero. In these simulations

$$t(x) = \begin{cases} 0 & \text{for } x < -0.1 \\ 15(x + 0.1) & \text{otherwise} \end{cases} \tag{4.22}$$

The masking field consisted of 336 cells arranged in 4 sections and 98 populations. Parameters were set at $A = B = 1$, $C = 0.2$, $D = 4$, $F = 24000$, $f_0 = 0.16$ and $g_0 = 1.0$. Unless otherwise noted, the learning rate was set at $e = 0.01$.

Finally, the initial bottom up connection weights are set so that $z_{ji} = 0.6$ for all values of i and j.

4.12 Summary

This chapter introduced a model of working memory and the Cohen-Grossberg masking field —the results of simulations supporting those in Cohen and Grossberg (1987) were presented. The masking field model was extended to cope with a larger input field and a larger chunk size, and the constraints which this places on the choice of parameters have been discussed. The necessity of placing the working memory and the masking field within an ART architecture was pointed out and I introduced a network, **MASKART**, in which this was achieved. A detailed account of the necessary modifications to both the masking field and the **ART3** algorithm was given.

In the next chapter, I shall describe a number of experiments employing the **MASKART** network, particularly those directed to the generation of plausible expectations in response to short input sequences.

Chapter 5

Application of the MASKART network

5.1 Experiments showing the effect of learning on masking field classifications and expectations

In the previous chapter I cited a passage from [69] which detailed how a large subset of a highly familiar pattern could elicit an expectation for the complete pattern. I pointed out that this notion of a pattern being "highly familiar", relative to the "familiarity" of any of its constituent sub-sequences, is not dependent on the relative frequency of occurrence of each (since a subsequence of a given sequence must be presented at least as *often* as the sequence itself) but is better thought of as being a statement about the "timing" details of the input pattern set. In this section, I will present the results of some simulations using **MASKART**, which will suggest a means of extracting information relating to expected continuations and will show how the nature of these expectations is dependent to some extent upon such timing details.

The simulations consisted of repeated presentation of a short sequence to the network. Before each presentation, all network activity was reset and the level of synaptic transmitter was set to a level corresponding to zero depletion. It was hoped that, after many presentations, the familiarity of the entire sequence would be reflected in a tendency for the $\mathcal{F}_2$ cell corresponding to that sequence to activate somewhat in advance of the completion of the sequence at $\mathcal{F}_0$, thus indicating the expected ending. It was unclear whether this anticipatory

activation of the larger $\mathcal{F}_2$ cell would be sustained throughout the period of time for which the pattern representing the near-complete sequence was active or whether the STM competition at $\mathcal{F}_2$ would eventually favour the activation of an $\mathcal{F}_2$ cell of the "correct" size (i.e. of size equal to the number of items active at $\mathcal{F}_1$). It was expected that the "rhythm" in which the sequence was presented would affect this issue.

In the first simulation the sequence $(1, 2, 3)$ was presented repeatedly to the network, each item having equal duration. To be explicit, for each presentation the pattern (1) was present in working memory for one unit of time, then the pattern $(1, 2)$ was activated at $\mathcal{F}_0$ (causing a transmitter reset) and was sustained there for 1 time unit, and finally, following a further transmitter reset, the pattern $(1, 2, 3)$ was present in working memory, also for 1 time unit. Throughout each presentation there was no external interference in the dynamics of the network, i.e. the activities of all cells in the network obeyed the differential equations previously specified. A measure of the network's expectations was then devised: at any point in time the instantaneous expectation was derived by subtracting the current pattern at level $1B$ (normalized to unit length) from the vector representing the current top-down pattern (also normalized to unit length). Both these patterns are assumed to be "available" at level $1A$. Negative expectations were set to zero. In this way significant activation of $\mathcal{F}_2$ cells representing supersets of the current working memory pattern would elicit positive expectations. Because it was expected that, in some circumstances, the activation of such cells would eventually be quenched by $\mathcal{F}_2$ competition, a further measure was devised. This involved integrating the expectation vector, as defined above, over the time since the presentation of the most recent item, and then normalizing the vector representing this integral. Note that this normalized-integral measure becomes equivalent to the former measure in cases where the anticipatory activation of an $\mathcal{F}_2$ cell survives the $\mathcal{F}_2$ competition , since the integral becomes dominated by the persistent expectation. Use of the normalized-integral measure has the advantage of allowing expectations due to transient activation of large $\mathcal{F}_2$ cells to be permanently registered.

On repeated presentation of the sequence $(1, 2, 3)$ the network quickly learns to expect the "2", and to a lesser extent the "3", in response to the first item and, likewise, the "3" in response to the first two items. This expectation results from the *transient* activation of the $\mathcal{F}_2$ cells that have learned to represent the patterns $(1, 2)$ and $(1, 2, 3)$. This pattern of behaviour

persists, even after 200 presentations of the sequence; at no stage does the masking field classify the current working memory pattern with a cell whose size differs from the number of items active in that pattern. The manner in which the masking field, after having been exposed to this input environment, proceeds to equilibrium for each of the patterns (1), $(1,2)$ and $(1,2,3)$, is shown in Figure 5.1.

A different result can be achieved by altering the rhythm of the input sequence. To illustrate this, the sequence $(1,2,3^2)$ was presented repeatedly to **MASKART**, where the superscript indicates that the last item had twice the duration of either of the other two items. The behaviour is at first similar to that shown in the previous simulation, namely that the integrated expectation measure shows plausible expectations without the activity of the cells responsible persisting at $\mathcal{F}_2$. However, after 36 sequence presentations the size-3 cell representing the whole sequence activates and remains active at $\mathcal{F}_2$ in response to presentation of the first two sequence items, eliciting *sustained* expectation for the item "3". Eventually the same $\mathcal{F}_2$ cell is able to activate enduringly in response to the presentation of the first item alone, thus eliciting expectations for the remaining two items (in the correct order). The manner in which the masking field proceeds to equilibrium in this case is shown in Figure 5.2.

As a result of the operational constraints stipulated previously, no subsequent input reset occurs and the cell which correctly represents the whole input sequence remains active throughout the remainder of the sequence presentation. Of course, if something unexpected occurs, e.g. a "4" occurs after the "2", then there will be an immediate transmitter reset.

The difference in behaviour exhibited in the second simulation can be attributed to the fact that the whole pattern $(1,2,3)$ is active in working memory for twice as long as either of the patterns (1) or $(1,2)$. This allows more learning of, in particular, bottom-up LTM traces to the $\mathcal{F}_2$ cell which comes to represent $(1,2,3)$, and this learning bias eventually becomes sufficient to tip the $\mathcal{F}_2$ competitive balance in favour of this cell, even in circumstances when its full pattern is not yet present at $\mathcal{F}_0$ and $\mathcal{F}_1$. In other words, the change in presentation rhythm has enhanced the "familiarity" of the whole sequence relative to its constituent sub-sequences.

A similar result is found for a pair of simulations in the first of which the sequence $(1,2,3,4)$ is repeatedly presented to the network in an isochronous rhythm, and in the second of which it is presented with the last item having twice the duration of each of the other items. Once again, in the first experiment *transient* activation of larger $\mathcal{F}_2$ cells leads to plausible expectations,

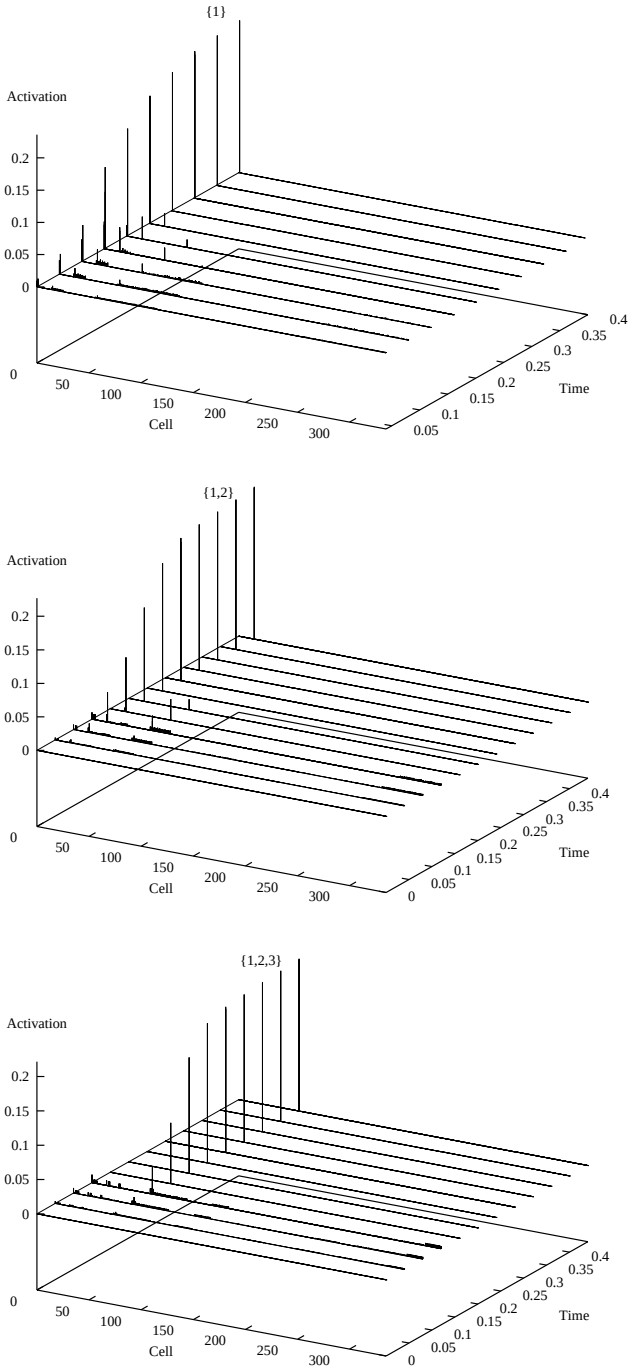


Figure 5.1: The masking field's response to the patterns (1), $(1,2)$ and $(1,2,3)$, after repeated presentation of the sequence $(1,2,3)$

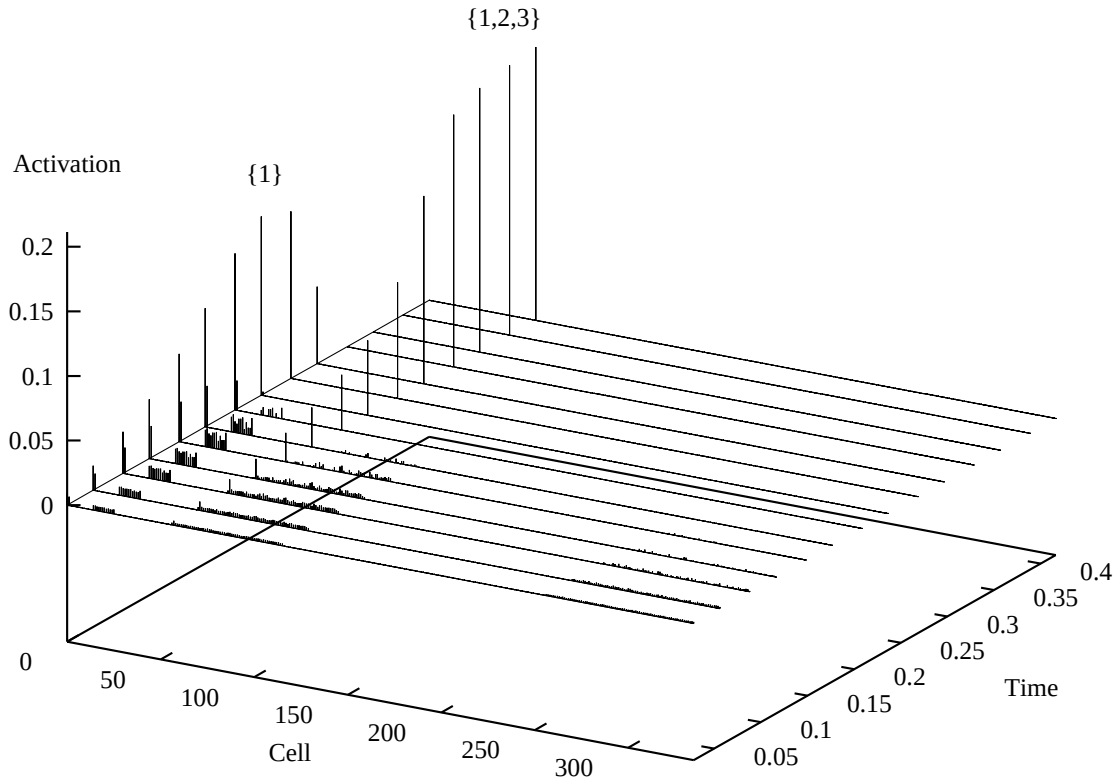


Figure 5.2: The masking field's response to the pattern (1), after repeated presentation of the sequence $(1, 2, 3^2)$

as indicated by the normalized-integral measure but, even after 200 presentations, only an $\mathcal{F}_2$ cell of size equal to the number of items in working memory is able to sustain its activity. By contrast, in the second experiment, after 29 presentations the $\mathcal{F}_2$ cell representing the pattern $(1, 2, 3, 4)$ activates enduringly in response to the pattern $(1, 2, 3)$; by the 50^{th} presentation the same cell is able to activate in response to the first item alone. In each case once enduringly activated the $\mathcal{F}_2$ cell remains active throughout the remainder of the sequence presentation.

Of course, the presentation of a short sequence in more than one context can inhibit the tendency of the network to generate *sustained* expectations in response to that sequence. To illustrate this, another simulation alternately presented **MASKART** with the sequences $(1, 2, 3^2)$ and $(1, 2, 4^2)$ (i.e. each in the non-isochronous rhythm used above). After 100 presentations of each, the following responses were given: presentation of the first item of a sequence, "1", elicited a large unsustained expectation for the item "2" and smaller but approximately equal expectations for "3" and "4"; presentation of the first two items of a sequence, "$(1, 2)$", elicited large, unsustained and approximately equal expectations for items "3" and "4". On full presentation of either of the sequences, the relevant $\mathcal{F}_2$ cell activated

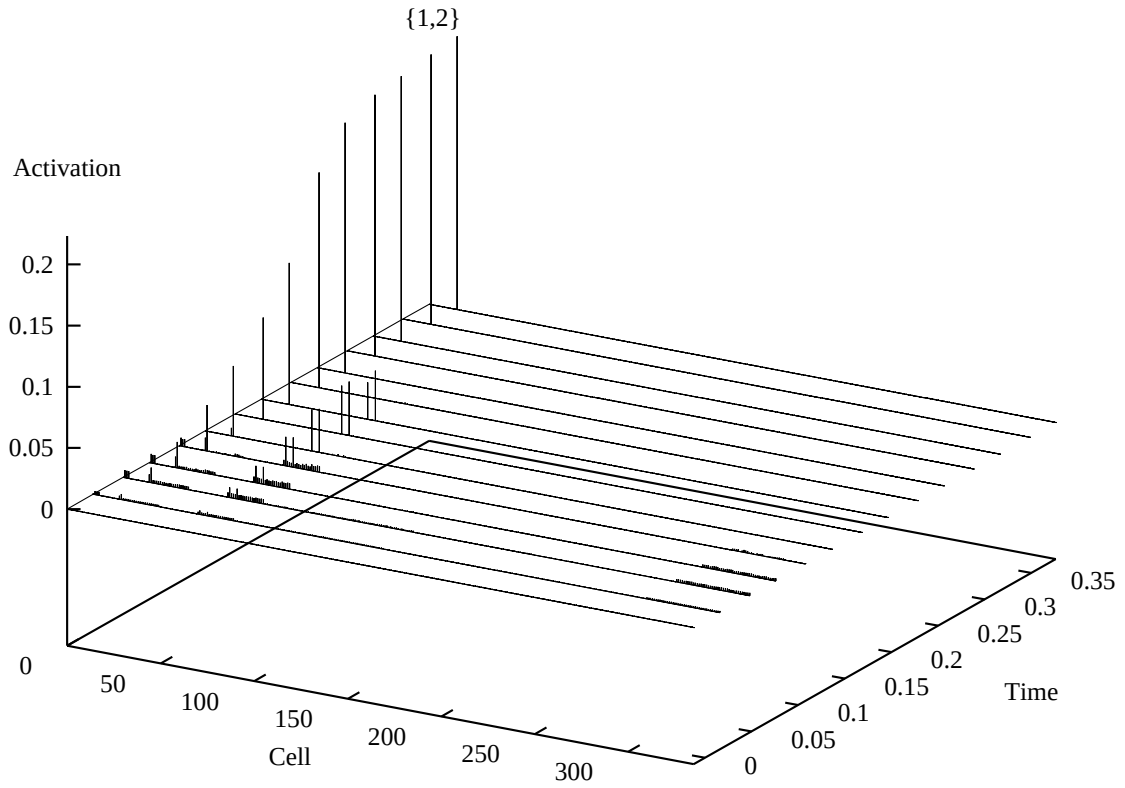


Figure 5.3: The masking field's response to the pattern $(1, 2)$, after repeated presentation of the sequences $(1, 2, 3^2)$ and $(1, 2, 4^2)$

leading to no sustained expectation. Thus the network has learned that a "3" or a "4" is likely to come after the short sequence "$(1, 2)$", with approximately equal probability. There is no *sustained* expectation for either, indicating that neither is exclusively expected. The response of the masking field to the item sequence $(1, 2)$ is shown in Figure 5.3.

Other "rhythmed" sequences were presented to **MASKART** and the network's behaviour in response to these sequences was confirmed as being "reasonable" in the light of the statistics of the input history.

The results presented in this section show how the repeated presentation of short sequences to the **MASKART** network can lead to the generation of expectations, when the network is presented with the initial parts of these sequences. The strength of these expectations and the way in which they are generated (i.e. whether they result from transient or sustained activation of the larger $\mathcal{F}_2$ cells), depends on the "familiarity" of the complete sequence relative to that of its constituent subsequences. The familiarity of a given (sub)sequence is a function of the amount of time that it spends in working memory—this can be affected both by the frequency with which the sequence occurs (possibly in multiple contexts) and the "rhythm"

with which it is presented. In this way the "statistics of the input environment" can be reflected in the pattern of expectations that **MASKART** generates. For instance, suppose that the sequence $(1,2)$ is presented often to **MASKART**, in several different contexts e.g. $(1,2,3)$, $(1,2,4)$, $(1,2,6)$. The increased familiarity of the sequence $(1,2)$, relative to that of each of the longer sequences, leads to a strengthening of the bottom-up input pathways to the $\mathcal{F}_2$ cell that comes to represent it. On presentation of the sequence $(1,2)$, therefore, that cell is able to compete strongly with the cells representing the longer sequences and thus reduces their ability to generate expectations (note that they also compete strongly with each other). By contrast, if the sequence $(1,2)$ always occurs as part of the sequence $(1,2,3)$, then the ability of the smaller $\mathcal{F}_2$ to compete with the larger cell will be reduced and the expectation of the continuation "3" will be correspondingly stronger. As has been seen, even when presented with such a situation the network does not respond by *sustaining* the activity of the larger cell on presentation of the sequence $(1,2)$. The network's reluctance to generate such sustained expectations is fortunate, since, once learned, these sustained expectations cannot be unlearned—this constitutes a serious problem for the current formulation (see Section 5.5.1).

The idea that expectation results from the partial activation of higher-order cells representing sequences that are familiar supersets of the current input, is markedly different from that suggested by the models described in Chapter 3 — in particular, the expectations generated in this chapter do not result from supervised learning and the dubious assumptions associated with such a regime. Bharucha [14] claims that the ART networks used by Gjerdingen [63] are "inescapably supervised" since they rely on information regarding "expectancy violation" for their *learning* — he refers to the ART reset event as indicating a mismatch between expectation and actuality. The point that Bharucha misses, however, is that, whereas in the supervised learning model it is this mismatch (signalled in advance by a supervisor) which *drives the learning process*, in ART-type networks such a mismatch *prevents* learning. The fact that an unfolding sequence can confirm or disconfirm expectations does not imply that the learning of those expectations was itself supervised.

The following section will outline the results of **MASKART** simulations which used simple musical data as input rather than arbitrarily chosen sequences.

5.2 Running MASKART with musical sequences as input

5.2.1 The input data set

As noted earlier, it was decided to use **MASKART** in the processing of data corresponding to musical sequences. It was hoped that **MASKART** would learn expectations based on typical melodic sub-sequences. The input set consisted of twelve simple nursery rhyme melodies. These melodies underwent simple preprocessing, so as to make them more appropriate to the constraints under which **MASKART** was operating. This simplification consisted of using only notes which fell on the "beat". This was beneficial for several reasons: firstly it allowed the input to the network to be restricted to representing the seven notes of the diatonic scale as opposed to the twelve chromatic notes—none of the nursery rhymes chosen had on-beat chromatic notes; secondly, the preprocessing allowed many instances of quick note repetitions (e.g fast trills) to be removed from the input set—**MASKART** is constrained by the "banana problem" and does not operate well in input environments rich in item repetition; thirdly, simplification allowed the duration of a note to be measured as an integral number of beats—this was invariably either one or two. Some latitude was allowed in the exact definition of a beat, for instance for melodies in 6:8 time a beat was generally held to occur every 3 quavers. In all cases, however, an attempt was made to equate the beat level with the (subjective) "tactus" of the tune. Interestingly, on performing this preprocessing it became rather difficult to select a set of ten well-known but different tunes, since several popular rhymes reduced to a single melody—for instance "Twinkle,Twinkle Little Star" and the "Alphabet Song" have essentially the same melodic structure. In one sense this was an encouraging observation in that it suggested that the beat level was a reasonable level at which to access an underlying melodic structure. In Chapter 8, I shall discuss further the idea of simplifying melodies in this way and will describe a mechanism which, by allowing a rhythmic variation in the attention paid to particular notes, could be used in the preprocessing stage.

The fact that the data refer to the notes of a set of melodies is transparent to the network. It merely sees the data as representing a sequence of items and, via long-term memory, extracts knowledge relating to regularities within that sequence. These regularities take the form of commonly occurring phrases or sub-sequences—the network learns "higher-level" representa-

tions of such sub-sequences and uses these in the generation of expectations corresponding to likely sequence continuations.

The data relating to the simplified nursery rhyme melodies are shown in Table 5.1. Each number refers to one of the diatonic scale degrees: *do* is numbered 1, *ti* is numbered 7. The underlining of a number denotes that the corresponding note is taken from an adjacent octave—in fact octave equivalence is assumed throughout this thesis. A superscript refers to the duration of the note in beats—where there is no superscript a duration of one beat is assumed. The spacing reflects the phrasing suggested both by the meter and, more importantly, by the verbal accompaniment. The reasons for including this extra information will be outlined in the next section. Table 5.2 contains data relating to the frequency of occurrence of each phrase, so defined, in the complete input set.

5.2.2 The evolution of working-memory patterns

The decision to present **MASKART** with data representing continuously presented sequences rather than just the short, isolated sequences used in the simulations above, forces some constraints on the evolution of the spatial pattern of activation at the working memory ($\mathcal{F}_0$). The principal constraint is that there should not be more than four items active at $\mathcal{F}_0$ at any time, since a working memory pattern that contradicted this constraint is almost bound to lead to a series of mismatch resets. For instance, suppose that the pattern $(1, 2, 3, 4)$ is active at $\mathcal{F}_0$ and that an $\mathcal{F}_2$ cell is active and is learning to represent this pattern. If we assume that we can have more than four items active at $\mathcal{F}_0$ then at the onset of the next item, say (5), there will be a mismatch reset which will bias the subsequent competition for activation at $\mathcal{F}_2$ against reactivating the previously active $\mathcal{F}_2$ cell. Since there are no size-5 cells in $\mathcal{F}_2$, the only way that a further reset can be avoided is by the activation of two or more $\mathcal{F}_2$ cells whose combined top-down expectation pattern is sufficiently similar to the current working memory pattern (or a superset of it) to avoid mismatch e.g. the two $\mathcal{F}_2$ cells representing the patterns $(1, 2, 3)$ and $(4, 5)$ may coactivate. However, for there to be no mismatch reset, the ratio of their activations and the degree of top-down learning that each has achieved must be precisely such that the combined top-down expectation pattern is highly similar to the current working memory pattern. Such a state of affairs is unlikely, making a series of mismatch resets very probable. Moreover, it may be that the most "appropriate" parsing of

Melody no.	Data
1	"Boys and Girls Come Out To Play" 5 4 5 1 2 4 5 1 5 4 5 1 2 4 5 1
2	"Rock-a-bye Baby" 1 3 2 4 3 4 2 1 1 3 2 4 3 4 2 1
3	"Three Blind Mice" $3\ 2\ 1^2$ $3\ 2\ 1^2$ $5\ 4\ 3^2$ $5\ 4\ 3^2$ $\underline{1}\ 7\ \underline{1}\ 5$ $\underline{1}\ 7\ \underline{1}\ 5$ $\underline{1}\ 7\ \underline{1}\ 5$ $3\ 2\ 1^2$
4	"Twinkle, Twinkle Little Star" ("The Alphabet Song") 1 5 6 5 4 3 2 1 5 4 3 2 5 4 3 2 1 5 6 5 4 3 2 1
5	"Here We Go Round The Mulberry Bush" ("Five Little Ducks" etc.) 1 1 5 1 2 2 $\underline{7}$ $\underline{5}$ 1 1 5 1 2 $\underline{5}$ 1^2
6	"Pop Goes The Weasel" 1 2 3 1 1 2 3 1 1 2 3 1 6 2 3 1
7	"Have You Seen The Little Ducks Play?" 1 3 5 5 4 2 3 1 1 3 5 5 4 2 3 1
8	"Polly Put The Kettle On" 5 5 3 1 2 2 $\underline{7}$ $\underline{5}$ 5 5 3 1 $\underline{6}$ $\underline{7}$ 1^2 3 4 3 1 2 2 $\underline{7}$ $\underline{5}$ 3 4 3 1 $\underline{6}$ $\underline{7}$ 1^2
9	"I'm a Little Teapot" 1 5 6 5 4 3 2 1 $\underline{1}$ 7 6 5 $\underline{1}$ 5 3 1
10	"Incy Wincy Spider" 1 3 2 1 3 5 4 3 6 5 4 3 1 3 2 1
11	"Little Bird" 5 3 4 2 1 3 5^2 5 3 4 2 1 5 1^2
12	"Frère Jacques" 1 2 3 1 1 2 3 1 3 4 5^2 3 4 5^2 5 5 3 1 5 5 3 1 1 $\underline{5}$ 1^2 1 $\underline{5}$ 1^2

Table 5.1: Melodies used as input to **MASKART**

Phrase	No.of Occurrences	Phrase	No. of Occurrences
1 1 5 1	2	3 4 2 1	2
1 2 3 1	5	3 4 3 1	2
1 3 2 1	2	$3\ 4\ 5^2$	2
1 3 2 4	2	3 5 4 3	1
1 3 5 5	2	4 2 3 1	2
$1\ 3\ 5^2$	1	4 3 2 1	3
$1\ 5\ 1^2$	3	5 3 4 2	2
1 5 3 1	1	$5\ 4\ 3^2$	2
1 5 6 5	3	5 4 3 2	2
1 7 1 5	3	5 4 5 1	2
1 7 6 5	1	5 5 3 1	4
2 2 7 5	3	6 2 3 1	1
2 4 5 1	2	6 5 4 3	1
$2\ 5\ 1^2$	1	$6\ 7\ 1^2$	2
$3\ 2\ 1^2$	3		

Table 5.2: Frequency of phrase occurrence in input set

the new working memory pattern is (1,2,3,4)(5) but that such a representation is inhibited by the original "input reset". It should be noted that Cohen and Grossberg's formulation of the masking field makes it very difficult for two $\mathcal{F}_2$ cells to coactivate strongly since all $\mathcal{F}_2$ cells inhibit all other $\mathcal{F}_2$ cells to some degree, even if the patterns that they represent do not overlap at all. In the simulations that were conducted to establish the infeasibility of such coactivation in **MASKART**, the inhibition between $\mathcal{F}_2$ cells that had no overlap of receptive fields was set to zero. This foreshadows work in later chapters.

Having established that the working memory pattern should not contain more than four active cells the question arises of how to implement this constraint. The basic idea is to reset the most active $\mathcal{F}_0$ cell(s) (i.e. the least recent items) when the working memory overloads. If this reset is restricted to the single most active $\mathcal{F}_0$ cell then the working memory comes to represent a "window" that always activates cells corresponding to the four most recent items (ignoring constraints imposed by item repetition). Such a representation has disadvantages, principally that (again in the absence of item repetition) only patterns consisting of four items will become active at $\mathcal{F}_0$ and thus be learned by the $\mathcal{F}_1 \rightarrow \mathcal{F}_2$ adaptive filter. It is possible to introduce a random element in the resetting of $\mathcal{F}_0$ cells as Nigrin [117] has done. This consists of resetting the most active $\mathcal{F}_0$ cell and then resetting the next most active cell with a given probability, say 0.5. Such a process results in the possibility of patterns consisting of three or four active items becoming active after $\mathcal{F}_0$ reset.

Simulations using this means of pruning the working memory pattern proved unsatisfactory primarily because the random element in the resetting of $\mathcal{F}_0$ was clumsy in the sense that it did not reflect, and neither did it claim to reflect, the specifically *melodic* nature of the input set. In other words, the random element in the reset process often resulted in a working memory pattern representing an "across phrase" group of notes. This led to there being little bias (other than that imposed by repetition constraints) to learn "within phrase" groups. This seemed unrealistic, in that the phrase structure of the nursery rhymes is made quite explicit, not only by metrical information (accents etc.) but also by their verbal accompaniment—in simple songs such as these, the verbal accompaniment is nearly always "in phase" with the metrical phrase structure (Gérard and Auxiette [60]). The phrase structure would be made even more explicit were the (unfamiliar) melodies to be presented to children "phrase by phrase" at the learning stage. To reflect this phrase structure, derived as it is from "external"

information, I employed the following means of resetting the working memory: whenever five items were activated in working memory, all items previous to the most recent "accented" note were reset. An accented note is defined as one that begins a metrical phrase and invariably also a linguistic segment in the verbal accompaniment. A further stipulation concerned the behaviour of the working memory in response to non-immediate repetitions: a non-immediate repetition caused reset of all items up to and including the first occurrence of the repeated item and, in addition, any items that remained in working memory but that occurred previous to the most recent accented note. In short, the idea is that, if the input sequence must be broken up at some point in order that working memory should not overload or contain repeated items, then the breaks should occur, all things being equal, at places indicated by metric and verbal information relating to the phrase structure.

It was realized that the introduction of, albeit very simple, phrasing information, not itself derived from the basic pitch and duration information, was in danger of diminishing the self-organizing nature of **MASKART**. In particular, the provision of this phrasing information largely negates **MASKART**'s rôle as a *parser* of input sequences, since much (though not all) of the parsing of the input sequences is accomplished by the mechanism for restricting the number of items in working memory. As noted previously, **MASKART**'s ability to indicate that a given sequence is best parsed into a number of *smaller* sequences is severely restricted by the high-vigilance regime in which it operates, which requires that the pattern of $\mathcal{F}_2$ activation at equilibrium is "non-distributed". As a result, the experiments described in the next section concentrate on the ability of **MASKART** to generate plausible expectations of phrase completion, which can perhaps be thought of as the "parsing" of the input sequence into a (number of) *larger*, familiar sequence(s). The decision to include the additional phrasing information was taken reluctantly but it seemed the only way to enable **MASKART** to operate effectively even under the considerable number of constraints imposed on it by its inherent simplicity. In later chapters many of these constraints will be relaxed on adoption of a more powerful, but more complex, formulation of the masking field, allowing the additional phrase information to be removed from the input set.

5.3 Experiments presenting MASKART with individual melodies

5.3.1 Procedure

The first set of experiments in which **MASKART** was presented with melody information was designed to test the ability of the network to function as a very simple sequential memory. To this end, each melody was presented repeatedly (10 times) to a **MASKART** network in which no previous learning had taken place. The melody was then presented once more to see if, after the occurrence of each note, the most expected note was the same as the actual next note. Thus **MASKART** was being tested for its ability to generate veridical expectations after exposure to a very restricted input set. The results are given in Tables 5.3—5.14. In each table the stimulus melody is shown at the top. In the leftmost column is the sequence that exists in the working memory as it changes with the presentation of each item—the evolution of this sequence is heavily influenced by the occurrence of repeated items and by the external phrasing information. The next columns contain information relating to the expectations generated by the working memory pattern: the two most expected items, together with a number representing the degree of expectation for each, are shown. The final column contains a symbol which summarizes the validity of the expectations generated: if the most expected note is "correct" then a "$*$" appears in this column; if the second most expected note is correct then a "$\circ$" appears; if the actual next note cannot possibly be expected, because it constitutes a repetition of a note already in working memory, a "—" appears; and if both expectations are incorrect then a "$\times$" appears. Where the most expected note is incorrect, a reason for the error is suggested (if one is readily apparent) in the form of a footnote. Where a footnote refers to, for instance, $M_{3,2,1}$, it means the masking field cell that has learned to respond to the sequence $(3, 2, 1)$.

MASKART's task is a comparatively simple one, nonetheless the ability to learn single sequences (e.g. melodies) after repeated isolated exposure is a prerequisite for a model of sequential memory. As will be seen, under certain circumstances **MASKART** is poor at performing the specified task and the identification of the reasons for this poor performance proved important in the development of later models.

Melody 1: 5 4 5 1 2 4 5 1 5 4 5 1 2 4 5 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
5	4	0.655	2	0.053	*
5,4	1	0.152	2	0.130	—
4,5	1	0.258	2	0.216	*
4,5,1	2	0.196	—	—	*
4,5,1,2	2	0.099	—	—	—
2,4	5	0.123	1	0.011	*
2,4,5	1	0.077	—	—	*
2,4,5,1	—	—	—	—	—
5	4	0.521	1	0.010	*
5,4	1	0.139	2	0.038	—
4,5	1	0.259	2	0.133	*
4,5,1	2	0.174	—	—	*
4,5,1,2	—	—	—	—	—
2,4	5	0.123	1	0.014	*
2,4,5	1	0.086	—	—	*
2,4,5,1	—	—	—	—	—

Table 5.3:

Melody 2: 1 3 2 4 3 4 2 1 1 3 2 4 3 4 2 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	3	0.347	2	0.023	*
1,3	2	0.140	4	0.088	*
1,3,2	4	0.194	—	—	*
1,3,2,4	—	—	—	—	—
3	4	0.215	1	0.028	*
3,4	2	0.141	1	0.023	*
3,4,2	1	0.122	—	—	*
3,4,2,1	—	—	—	—	—
1	3	0.299	2	0.015	*
1,3	2	0.126	4	0.052	*
1,3,2	4	0.146	—	—	*
1,3,2,4	—	—	—	—	—
3	4	0.216	1	0.030	*
3,4	2	0.140	1	0.024	*
3,4,2	1	0.129	—	—	*
3,4,2,1	—	—	—	—	—

Table 5.4:

Melody 3: 3 2 1^2 3 2 1^2 5 4 3^2 5 4 3^2 $\underline{1}$ 7 $\underline{1}$ 5 $\underline{1}$ 7 $\underline{1}$ 5 $\underline{1}$ 7 $\underline{1}$ 5 3 2 1^2					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
3	2	0.664	5	0.231	$*$
3,2	1	1.127	5	0.008	$*$
3,2,1	5	0.018	7	0.012	—
3	5	0.301	2	0.253	$\circ^a$
3,2	1	0.581	5	0.013	$*$
3,2,1	5	0.024	7	0.013	$*$
3,2,1,5	4	0.065	7	0.029	$*$
5,4	3	0.388	1	0.015	$*$
5,4,3	1	0.026	2	0.002	—
5	4	0.347	7	0.064	$*$
5,4	3	0.564	1	0.023	$*$
5,4,3	1	0.033	2	0.008	$*$
5,4,3,1	2	0.041	7	0.029	$\circ$
1,7	5	0.100	3	0.014	—
7,1	5	0.226	3	0.009	$*$
7,1,5	3	0.020	4	0.002	—
1	7	0.949	3	0.318	$*$
1,7	5	0.071	3	0.008	—
7,1	5	0.159	3	0.008	$*$
7,1,5	3	0.022	4	0.004	—
1	7	2.814	3	0.329	$*$
1,7	5	0.103	3	0.009	—
7,1	5	0.171	3	0.009	$*$
7,1,5	3	0.022	4	0.005	$*$
7,1,5,3	2	0.115	4	0.037	$*$
3,2	1	0.850	5	0.015	$*$
3,2,1	5	0.019	7	0.004	—

[a] Bias against $M_{3,2,1}$ due to previous reset.

Table 5.5:

Melody 4: 1 5 6 5 4 3 2 1 5 4 3 2 5 4 3 2 1 5 6 5 4 3 2 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	5	0.191	6	0.051	$*$
1 5	6	0.103	4	0.015	$*$
1 5 6	4	0.021	3	0.006	—
6 5	4	0.111	3	0.039	$*$
6 5 4	3	0.104	1	0.005	$*$
6 5 4 3	2	0.016	1	0.008	$*$
4 3 2	5	0.131	1	0.056	$\circ^a$
4 3 2 1	5	0.081	6	0.001	$*$
5	1	1.28	4	0.143	$\circ^b$
5 4	6	0.285	3	0.107	$\circ^c$
5 4 3	6	0.126	2	0.066	$\circ$
5 4 3 2	6	0.068	1	0.014	—
5	1	1.51	6	0.191	$\times$
5 4	6	0.650	3	0.100	$\circ$
5 4 3	6	0.138	2	0.107	$\circ$
5 4 3 2	6	0.066	1	0.021	$\circ$
1	5	0.199	6	0.053	$*$
1 5	6	0.114	4	0.007	$*$
1 5 6	4	0.021	3	0.006	—
6 5	4	0.115	3	0.041	$*$
6 5 4	3	0.109	1	0.005	$*$
6 5 4 3	2	0.017	1	0.009	$*$
4 3 2	5	0.139	1	0.059	$\circ^d$
4 3 2 1	5	0.089	6	0.001	—

[a] $M_{5,4,3,2}$ dominates

[b] $M_{1,5}$ dominates

[c] $M_{6,5,4,3}$ dominates

[d] $M_{5,4,3,2}$ dominates

Table 5.6:

Melody 5: 1 1 5 1 2 2 $\underline{7}$ $\underline{5}$ 1 1 5 1 2 $\underline{5}$ 1^2					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	5	0.447	2	0.090	$*$
1,5	2	0.537	5	0.032	—
5,1	2	0.942	1	0.077	$*$
5,1,2	7	0.082	—	—	$*$
5,1,2,7	—	—	—	—	—
2,7,5	1	0.143	—	—	$*$
1	5	0.015	1	0.013	$*$
1,5	2	0.355	5	0.060	—
5,1	2	4.922	7	0.022	$*$
5,1,2	—	—	—	—	—
2,5	1	0.293	7	0.118	$*$
2,5,1	7	0.132	—	—	—

Table 5.7:

Melody 6: 1 2 3 1 1 2 3 1 1 2 3 1 6 2 3 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	2	0.439	3	0.034	$*$
1,2	3	0.596	6	0.024	$*$
1,2,3	6	0.044	1	0.004	—
2,3,1	6	0.071	—	—	—
1	2	0.054	3	0.008	$*$
1,2	3	0.212	6	0.030	$*$
1,2,3	6	0.053	—	—	—
2,3,1	6	0.069	—	—	—
1	2	0.050	3	0.008	$*$
1,2	3	0.214	6	0.027	$*$
1,2,3	6	0.052	—	—	—
2,3,1	6	0.068	—	—	$*$
2,3,1,6	—	—	—	—	—
6,2	3	0.119	1	0.017	$*$
6,2,3	1	0.113	—	—	$*$
6,2,3,1	—	—	—	—	—

Table 5.8:

Melody 7: 1 3 5 5 4 2 3 1 1 3 5 5 4 2 3 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	3	0.492	5	0.156	*
1,3	5	0.774	4	0.045	*
1,3,5	4	0.045	2	0.002	*
1,3,5,4	2	0.030	—	—	*
4,2	3	0.199	1	0.030	*
4,2,3	1	0.087	5	0.001	*
4,2,3,1	5	0.012	—	—	—
1	3	0.382	5	0.082	*
1,3	5	0.697	4	0.032	*
1,3,5	4	0.044	2	0.001	*
1,3,5,4	2	0.025	—	—	*
4,2	3	0.183	1	0.029	*
4,2,3	1	0.090	5	0.001	*
4,2,3,1	5	0.013	—	—	—

Table 5.9:

Melody 8: 5 5 3 1 2 2 $\underline{7}$ $\underline{5}$ 5 5 3 1 $\underline{6}$ $\underline{7}$ 1^2 3 4 3 1 2 2 $\underline{7}$ $\underline{5}$ 3 4 3 1 $\underline{6}$ $\underline{7}$ 1^2					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
5	3	0.065	2	0.038	$*$
5 3	1	0.122	4	0.044	$*$
5 3 1	2	0.166	4	0.055	$*$
5 3 1 2	4	0.065	7	0.030	$\circ^a$
2 7	5	0.076	3	0.009	$*$
2 7 5	3	0.030	6	0.003	—
5	3	0.057	1	0.010	$*$
5 3	1	0.120	4	0.045	$*$
5 3 1	2	0.173	4	0.055	$\times^b$
5 3 1 6	7	0.182	2	0.151	$*$
6 7	1	5.520	2	0.006	$*^c$
6 7 1	—	—	—	—	—
6 7 1 3	4	0.024	5	0.012	$*$
3 4	1	0.094	5	0.027	—
4 3	1	0.192	2	0.036	$*$
4 3 1	2	0.117	5	0.041	$*$
4 3 1 2	5	0.062	6	0.006	$\times^d$
2 7	5	0.080	3	0.085	$*$
2 7 5	3	0.038	1	0.009	$*$
2 7 5 3	1	0.111	4	0.001	$\circ^e$
3 4	1	0.072	5	0.029	—
4 3	1	0.162	2	0.029	$*$
4 3 1	2	0.118	5	0.044	$\times^f$
4 3 1 6	7	0.234	2	0.081	$*$
6 7	1	5.603	2	0.006	$*$
6 7 1	—	—	—	—	—

[a] $M_{4,3,1,2}$ dominates

[b] $M_{5,3,1,2}$ dominates

[c] Sustained expectation

[d] $M_{5,3,1,2}$ dominates

[e] $M_{5,3,1}$ dominates

[f] $M_{4,3,1,2}$ dominates

Table 5.10:

Melody 9: 1 5 6 5 4 3 2 1 $\underline{1}$ 7 6 5 $\underline{1}$ 5 3 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	5	0.150	7	0.065	$*$
1,5	3	0.107	6	0.054	$\circ^a$
1,5,6	7	0.100	3	0.026	—
6,5	4	0.057	1	0.025	$*$
6,5,4	3	0.050	1	0.006	$*$
6,5,4,3	1	0.012	2	0.003	$\circ$
4,3,2	1	0.029	—	—	$*$
4,3,2,1	5	0.005	—	—	—
1	5	0.146	7	0.067	$\circ^b$
1,7	6	0.037	5	0.007	$*$
1,7,6	5	0.053	—	—	$*$
1,7,6,5	3	0.007	4	0.001	—
1	5	0.187	6	0.020	$*$
1,5	3	0.104	6	0.049	$*$
1,5,3	6	0.013	4	0.004	—
5,3,1	6	0.015	4	0.004	—

[a] $M_{1,5,3}$ dominates

[b] $M_{1,5,6}$ dominates

Table 5.11:

Melody 10: 1 3 2 1 3 5 4 3 6 5 4 3 1 3 2 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	3	0.334	2	0.021	*
1,3	2	0.339	5	0.008	*
1,3,2	5	0.001	4	0.001	—
3,2,1	5	0.002	4	0.001	—
3	5	0.114	1	0.038	*
3,5	4	0.120	6	0.023	*
3,5,4	6	0.067	1	0.002	—
5,4,3	6	0.103	1	0.001	*
5,4,3,6	1	0.001	2	0.001	—
6,5	4	0.107	3	0.016	*
6,5,4	3	0.068	—	—	*
6,5,4,3	1	0.001	2	0.001	*
1	3	0.341	2	0.022	*
1,3	2	0.337	5	0.006	*
1,3,2	5	0.001	4	0.001	—
3,2,1	5	0.002	4	0.001	—

Table 5.12:

Melody 11: 5 3 4 2 1 3 5^2 5 3 4 2 1 5 1^2					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
5	1	0.370	3	0.264	$\circ^a$
5,3	1	0.290	4	0.086	$\circ^b$
5,3,4	2	0.106	1	0.086	$*$
5,3,4,2	1	0.067	—	—	$*$
1	5	0.267	3	0.141	$\circ^c$
1,3	5	0.276	4	0.001	$*$
1,3,5	4	0.008	2	0.001	—
5	1	0.234	3	0.154	$\circ$
5,3	4	0.094	1	0.031	$*$
5,3,4	2	0.111	1	0.045	$*$
5,3,4,2	1	0.060	—	—	$*$
1	5	0.259	3	0.108	$*$
1,5	3	0.553	4	0.001	—
5,1	3	0.982	4	0.001	—

[a] $M_{5,1}$ and $M_{1,5}$ dominate

[b] $M_{1,3,5}$ dominates

[c] $M_{1,5}$ and $M_{5,1}$ dominate

Table 5.13:

Melody 12: 1 2 3 1 1 2 3 1 3 4 5^2 3 4 5^2 5 5 3 1 5 5 3 1 1 $\underline{5}$ 1^2 1 $\underline{5}$ 1^2					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	5	7.58	2	0.180	○[a]
1 2	3	0.385	5	0.006	∗
1 2 3	5	0.050	4	0.002	—
2 3 1	5	0.040	4	0.002	—
1	5	6.745	2	0.124	○
1 2	3	0.111	5	0.007	∗
1 2 3	5	0.047	4	0.002	—
2 3 1	5	0.263	4	0.003	—
3	5	1.015	4	0.243	○
3 4	5	0.600	1	0.002	∗
3 4 5	1	0.022	2	0.001	—
3	5	1.127	4	0.120	○[b]
3 4	5	0.548	1	0.002	∗
3 4 5	1	0.024	2	0.001	—
5	1	0.398	3	0.099	○
5 3	4	0.312	1	0.088	○[c]
5 3 1	4	0.078	2	0.042	—
5	1	0.501	3	0.102	○
5 3	4	0.377	1	0.083	○
5 3 1	4	0.089	2	0.042	—
1	5	5.600	2	0.241	∗
1 5	3	0.086	4	0.005	—
5 1	3	0.165	1	0.013	—
1	2	0.314	5	0.125	○[d]
1 5	3	0.109	4	0.007	—
5 1	3	0.109	1	0.036	—

[a] $M_{5,1}$ and $M_{1,5}$ dominate

[b] $M_{5,3}$ dominates

[c] $M_{3,4,5}$ dominates

[d] $M_{1,2}$ dominates

Table 5.14:

5.3.2 Overview of errors

MASKART's errors in performing the task described above can be divided into several categories. An overview of these error categories is given here.

- **MASKART** is not good at expecting phrase repetitions. This is illustrated, for instance, by its response to the fourth event in Melody 1. When the "3" occurs, it causes a reset event, which biases the masking field against the immediate reactivation of previously active cells. In this case, $M_{3,2,1}$ is thus disadvantaged and is therefore unable to "contribute" to the correct expectation of a "2". In addition, the cell $M_{3,2}$ has also undergone a less recent reset and therefore contributes less than it would otherwise have done. A similar situation will always occur when the same phrase occurs twice (or more) in succession.

- A very common error is illustrated by the network's response to the seventh event in Melody 2. The network has already learned the phrases $(4,3,2,1)$ and $(5,4,3,2)$, each of which occurs twice in the melody. On presentation of the subsequence $(4,3,2)$ both cells will receive significant input, the former receiving slightly more. In spite of this, the "5" is more expected than the "1". This is partly because the top-down weight from $M_{5,4,3,2}$ to the cell corresponding to item "5" is greater than the top-down weight from $M_{4,3,2,1}$ to item "1". In addition, the "5" receives additional "support" from the transient activation of cell $M_{5,4,3}$ and $M_{5,4}$. This type of error is difficult to prevent and is discussed in greater detail in the next chapter.

- Errors sometimes occur when four items are active in working memory. This is primarily because the next item is bound to cause an reset event, whether due to overload or item repetition. A four-item phrase can thus never be active in working memory at the same time as the item that follows it and expectations cannot be learned. The expectations resulting from four-item phrases are often small, partly due to the strong competition of the relevant size-4 cell, and partly due to the restricted range of items which can be expected at all.

- Errors sometimes occur when there is only a single item active in working memory — this reflects the paucity of context information avalailable for generating expectations. This error is particularly common in situations where the network has learned a number

of "isolated" two-item phrases. For instance, in Melody 6 the item repetitions in the last two phrases lead to the learning of the subphrases $(5, 1)$ and $(1, 5)$. This results in a situation in which "5" is strongly expected in response to "1" and vice versa. Ironically, on presentation of the first item in the last phrase, the correct expectation for "5" is not elicited due to recent reset.

5.3.3 A second experiment with altered input

The last of these points suggested a further experiment. The learning of isolated two-item sequences is often the result of in-phrase item repetition — the phrases are "split" to accomodate the restrictions placed on the evolution of working memory patterns by the "banana problem". In Chapter 7, this problem is addressed fully, so that item repetitions are not treated differently from other events; this requires a considerable increase in network complexity. To foreshadow this modification, I decided to produce a second set of "melodies" in which there were no in-phrase repetitions. This was achieved by replacing each repetition found in the original phrases with a substitute note. The substitute was chosen using a number of heuristics: a particular replacement is chosen if the resulting phrase is one that already appears elsewhere in the input set; a replacement note is chosen so that we do not end up with too many "anagramatical" phrases — since there is a restricted number of cells per masking field population, **MASKART** is incapable of learning a large number of phrases consisting of different orderings of the same items; when the repeated note is of long duration (2 beats), the repetition is prevented by replacing the shorter occurrence. Note that no attempt was made to ensure that the resulting sequences were melodically plausible. This experiment was simply to see how **MASKART** coped with a data set that contained no in-phrase repetitions. The transformations are shown in Table 5.15, the resulting sequences are shown in Table 5.16 and the numbers of each phrase are shown in Table 5.17.

Each of the altered "melodies" was presented 10 times to the **MASKART** network and its expectations were assessed as before. The results are given in Tables 5.18—5.29. It should be noted that the removal of all in-phrase repetitions from the input set also clarifies the phrase structure of the sequences — all $\mathcal{F}_0$ reset events cause all notes previous to the most recent "accented" note to be removed from working memory.

From	To
1 1 5 1	1 2 5 7
1 2 3 1	1 2 3 7
1 3 2 1	1 3 2 4
1 3 5 5	1 3 5^2
1 5 1^2	6 5 1^2
1 5 3 1	1 5 3 6
1 5 6 5	1 5 6 7
1 7 1 5	1 7 6 5
2 2 7 5	2 6 7 5
3 4 3 1	3 4 2 1
3 5 4 3	3 5 4 6
5 5 3 1	5 6 3 1
5 4 5 1	5 4 6 1

Table 5.15: Transformation of phrases to avoid in-phrase repetition

Melody no.	Data
1b	5 4 6 1 2 4 5 1 5 4 6 1 2 4 5 1
2b	1 3 2 4 3 4 2 1 1 3 2 4 3 4 2 1
3b	$3\ 2\ 1^2$ $3\ 2\ 1^2$ $5\ 4\ 3^2$ $5\ 4\ 3^2$ 1 7 6 5 1 7 6 5 1 7 6 5 $3\ 2\ 1^2$
4b	1 5 6 7 4 3 2 1 5 4 3 2 5 4 3 2 1 5 6 7 4 3 2 1
5b	1 2 5 7 2 6 7 5 1 2 5 7 $2\ 5\ 1^2$
6b	1 2 3 7 1 2 3 7 1 2 3 7 6 2 3 1
7b	$1\ 3\ 5^2$ 4 2 3 1 $1\ 3\ 5^2$ 4 2 3 1
8b	5 6 3 1 2 6 7 5 5 6 3 1 $6\ 7\ 1^2$ 3 4 2 1 2 6 7 5 3 4 2 1 $6\ 7\ 1^2$
9b	1 5 6 7 4 3 2 1 1 7 6 5 1 5 3 6
10b	1 3 2 4 3 5 4 6 6 5 4 3 1 3 2 4
11b	5 3 4 2 $1\ 3\ 5^2$ 5 3 4 2 $6\ 5\ 1^2$
12b	1 2 3 7 1 2 3 7 $3\ 4\ 5^2$ $3\ 4\ 5^2$ 5 6 3 1 5 6 3 1 $6\ 5\ 1^2$ $6\ 5\ 1^2$

Table 5.16: Altered "melodies"

Phrase	No.of Occurrences	Phrase	No. of Occurrences
1 2 3 7	5	3 5 4 6	1
1 2 5 7	2	4 2 3 1	2
1 3 2 4	4	4 3 2 1	3
1 3 5^2	3	5 3 4 2	2
1 5 3 6	1	5 4 3^2	2
1 5 6 7	3	5 4 3 2	2
1 7 6 5	4	5 4 6 1	2
2 4 5 1	2	5 6 3 1	4
2 5 1^2	1	6 2 3 1	1
2 6 7 5	3	6 5 1^2	3
3 2 1^2	3	6 5 4 3	1
3 4 2 1	4	6 7 1^2	2
3 4 5^2	2		

Table 5.17: Frequency of phrase occurrence in altered input set

Melody 1b: 5 4 6 1 2 4 5 1 5 4 6 1 2 4 5 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
5	4	0.177	2	0.057	$*$
5,4	2	0.159	6	0.089	$\circ^a$
5,4,6	1	0.072	2	0.035	$*$
5,4,6,1	2	0.082	1	0.000	$*$
2	4	0.309	5	0.027	$*$
2,4	5	0.154	1	0.014	$*$
2,4,5	1	0.044	6	0.005	$*$
2,4,5,1	6	0.019	—	—	—
5	4	0.125	6	0.017	$*$
5,4	6	0.096	2	0.033	$*$
5,4,6	1	0.074	2	0.019	$*$
5,4,6,1	2	0.062	1	0.000	$*$
2	4	0.287	5	0.022	$*$
2,4	5	0.156	1	0.014	$*$
2,4,5	1	0.045	6	0.006	$*$
2,4,5,1	6	0.021	—	—	—

[a] $M_{2,4,5}$ dominates

Table 5.18:

Melody 2: 1 3 2 4 3 4 2 1 1 3 2 4 3 4 2 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	3	0.347	2	0.023	*
1,3	2	0.140	4	0.088	*
1,3,2	4	0.194	—	—	*
1,3,2,4	—	—	—	—	—
3	4	0.215	1	0.028	*
3,4	2	0.141	1	0.023	*
3,4,2	1	0.122	—	—	*
3,4,2,1	—	—	—	—	—
1	3	0.299	2	0.015	*
1,3	2	0.126	4	0.052	*
1,3,2	4	0.146	—	—	*
1,3,2,4	—	—	—	—	—
3	4	0.216	1	0.030	*
3,4	2	0.140	1	0.024	*
3,4,2	1	0.129	—	—	*
3,4,2,1	—	—	—	—	—

Table 5.19:

Melody 3b: 3 2 1^2 3 2 1^2 5 4 3^2 5 4 3^2 1 7 6 5 1 7 6 5 1 7 6 5 3 2 1^2					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
3	2	0.409	5	0.168	$*$
3,2	1	0.963	5	0.008	$*$
3,2,1	5	0.015	7	0.003	—
3	5	0.244	2	0.238	$\circ^a$
3,2	1	0.631	5	0.013	$*$
3,2,1	5	0.021	7	0.003	$*$
3,2,1,5	4	0.065	7	0.002	$*$
5,4	3	0.390	1	0.014	$*$
5,4,3	1	0.025	2	0.002	—
5	4	0.347	3	0.029	$*$
5,4	3	0.566	1	0.023	$*$
5,4,3	1	0.031	2	0.007	$*$
5,4,3,1	2	0.041	7	0.029	$\circ^b$
1,7	6	0.113	5	0.016	$*$
1,7,6	5	0.068	3	0.003	$*$
1,7,6,5	3	0.002	2	0.001	—
1	3	0.370	2	0.307	$\times^c$
1,7	6	0.087	5	0.008	$*$
1,7,6	5	0.058	3	0.001	$*$
1,7,6,5	3	0.001	2	0.001	—
1	3	0.402	2	0.333	$\times$
1,7	6	0.089	5	0.008	$*$
1,7,6	5	0.051	3	0.001	$*$
1,7,6,5	3	0.002	2	0.001	$*$
3	2	0.429	5	0.184	$*$
3,2	1	1.024	5	0.008	$*$
3,2,1	5	0.016	7	0.002	—

[a] Bias against $M_{3,2,1}$ due to previous reset.

[b] $M_{3,2,1}$ dominates

[c] Bias against $M_{1,7,6,5}$ due to previous reset, $M_{3,2,1}$ dominates

Table 5.20:

Melody 4b: 1 5 6 7 4 3 2 1 5 4 3 2 5 4 3 2 1 5 6 7 4 3 2 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	5	0.196	6	0.055	$*$
1 5	6	0.123	7	0.030	$*$
1 5 6	7	0.107	4	0.002	$*$
1 5 6 7	4	0.001	—	—	$*$
4	5	0.482	3	0.310	$\circ^a$
4 3	5	0.234	2	0.138	$\circ$
4 3 2	5	0.129	1	0.058	$\circ$
4 3 2 1	5	0.075	—	—	$*$
5	1	1.09	4	0.141	$\circ^b$
5 4	3	0.111	2	0.016	$*$
5 4 3	2	0.059	1	0.007	$*$
5 4 3 2	1	0.014	—	—	—
5	1	1.52	4	0.071	$\circ$
5 4	3	0.133	2	0.011	$*$
5 4 3	2	0.071	1	0.008	$*$
5 4 3 2	1	0.020	—	—	$*$
1	5	0.204	6	0.057	$*$
1 5	6	0.141	7	0.032	$*$
1 5 6	7	0.108	4	0.001	$*$
1 5 6 7	4	0.001	—	—	$*$
4	5	0.466	3	0.317	$\circ$
4 3	5	0.241	2	0.144	$\circ$
4 3 2	5	0.139	1	0.059	$\circ$
4 3 2 1	5	0.083	—	—	—

[a] $M_{5,4}$, $M_{5,4,3}$ and $M_{5,4,3,2}$ dominate

[b] $M_{1,5}$ dominates

Table 5.21:

Melody 5b: 1 2 5 7 2 6 7 5 1 2 5 7 2 5 1^2					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	2	0.372	5	0.101	$*$
1,2	5	0.424	7	0.029	$*$
1,2,5	7	0.138	6	0.003	$*$
1,2,5,7	6	0.011	—	—	—
2	5	0.200	6	0.146	$\circ^a$
2,6	7	0.078	5	0.015	$*$
2,6,7	5	0.050	1	0.005	$*$
2,6,7,5	1	0.049	—	—	$*$
1	2	0.344	5	0.091	$*$
1,2	5	0.426	7	0.032	$*$
1,2,5	7	0.144	6	0.002	$*$
1,2,5,7	6	0.009	—	—	—
2	5	0.191	6	0.156	$*$
2,5	1	0.411	6	0.016	$*$
2,5,1	7	0.043	1	0.008	—

[a] $M_{2,5}$ and $M_{2,5,1}$ dominate

Table 5.22:

Melody 6b: 1 2 3 7 1 2 3 7 1 2 3 7 6 2 3 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	2	0.433	3	0.027	$*$
1,2	3	0.169	6	0.031	$*$
1,2,3	7	0.105	6	0.036	$*$
1,2,3,7	6	0.014	—	—	—
1	2	0.067	3	0.009	$*$
1,2	3	0.096	6	0.040	$*$
1,2,3	7	0.082	6	0.045	$*$
1,2,3,7	6	0.014	—	—	—
1	2	0.076	3	0.015	$*$
1,2	3	0.138	6	0.035	$*$
1,2,3	7	0.075	6	0.039	$*$
1,2,3,7	6	0.015	—	—	$*$
6	2	0.091	3	0.022	$*$
6,2	3	0.066	1	0.027	$*$
6,2,3	1	0.089	7	0.002	$*$
6,2,3,1	7	0.024	—	—	—

Table 5.23:

Melody 7: 1 3 5 5 4 2 3 1 1 3 5 5 4 2 3 1					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	3	0.492	5	0.156	*
1,3	5	0.774	4	0.045	*
1,3,5	4	0.045	2	0.002	*
1,3,5,4	2	0.030	—	—	*
4,2	3	0.199	1	0.030	*
4,2,3	1	0.087	5	0.001	*
4,2,3,1	5	0.012	—	—	—
1	3	0.382	5	0.082	*
1,3	5	0.697	4	0.032	*
1,3,5	4	0.044	2	0.001	*
1,3,5,4	2	0.025	—	—	*
4,2	3	0.183	1	0.029	*
4,2,3	1	0.090	5	0.001	*
4,2,3,1	5	0.013	—	—	—

Table 5.24:

Melody 8b: 5 6 3 1 2 6 7 5 5 6 3 1 6 7 1^2 3 4 2 1 2 6 7 5 3 4 2 1 6 7 1^2					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
5	6	0.149	3	0.035	$*$
5 6	3	0.083	7	0.044	$*$
5 6 3	1	0.062	7	0.007	$*$
5 6 3 1	7	0.090	2	0.003	$\circ^a$
2	6	0.355	3	0.044	$*$
2 6	7	0.177	5	0.052	$*$
2 6 7	1	0.192	5	0.083	$\circ^b$
2 6 7 5	1	0.085	3	0.003	—
5	6	0.132	3	0.028	$*$
5 6	3	0.090	7	0.024	$*$
5 6 3	1	0.064	7	0.005	$*$
5 6 3 1	7	0.086	2	0.003	—
6	7	0.347	2	0.223	$*$
6 7	1	0.719	2	0.119	$*$
6 7 1	2	0.033	3	0.018	$\circ$
6 7 1 3	2	0.085	5	0.060	$\times$
3 4	2	0.129	1	0.017	$*$
3 4 2	1	0.055	6	0.003	$*$
3 4 2 1	6	0.002	5	0.001	—
2	6	0.361	7	0.020	$*$
2 6	7	0.181	5	0.149	$*$
2 6 7	1	0.187	5	0.090	$\circ$
2 6 7 5	1	0.085	3	0.005	$\circ$
3	4	0.327	5	0.108	$*$
3 4	2	0.137	1	0.014	$*$
3 4 2	1	0.056	6	0.003	$*$
3 4 2 1	6	0.003	5	0.002	$*$
6	5	1.900	7	0.318	$\circ$
6 7	1	0.770	2	0.204	$*$
6 7 1	2	0.034	3	0.021	—

[a] $M_{6,7,1}$ dominates

[b] $M_{6,7,1}$ dominates

Table 5.25:

Melody 9b: 1 5 6 7 4 3 2 1 1 7 6 5 1 5 3 6					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	5	0.139	7	0.063	$*$
1,5	6	0.073	3	0.040	$*$
1,5,6	7	0.187	3	0.051	$*$
1,5,6,7	3	0.032	4	0.001	$\circ$
4	3	0.203	2	0.018	$*$
4,3	2	0.071	1	0.012	$*$
4,3,2	1	0.029	5	0.001	$*$
4,3,2,1	5	0.004	6	0.001	—
1	5	0.142	7	0.062	$\circ^a$
1,7	6	0.049	5	0.021	$*$
1,7,6	5	0.128	3	0.001	$*$
1,7,6,5	3	0.015	4	0.001	—
1	5	0.144	7	0.034	$*$
1,5	6	0.055	3	0.040	$\circ^b$
1,5,3	6	0.050	7	0.004	$*$
1,5,3,6	7	0.053	4	0.001	—

[a] $M_{1,5}$ dominates

[b] $M_{1,5,6}$ dominates

Table 5.26:

Melody 10b: 1 3 2 4 3 5 4 6 6 5 4 3 1 3 2 4					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	3	0.331	2	0.019	$*$
1,3	2	0.099	4	0.019	$*$
1,3,2	4	0.072	5	0.001	$*$
1,3,2,4	5	0.004	6	0.001	—
3	5	0.102	1	0.032	$*$
3,5	6	0.086	4	0.071	$\circ^a$
3,5,4	6	0.145	1	0.004	$*$
3,5,4,6	6	0.126	1	0.003	—
6	5	0.063	4	0.012	$*$
6,5	4	0.051	3	0.040	$*$
6,5,4	3	0.109	1	0.000	$*$
6,5,4,3	3	0.167	1	0.001	$\circ$
1	3	0.339	2	0.020	$*$
1,3	2	0.106	4	0.020	$*$
1,3,2	4	0.076	5	0.001	$*$
1,3,2,4	5	0.004	6	0.001	—

[a] $M_{6,5,4,3}$ contributes

Table 5.27:

Melody 11b: 5 3 4 2 1 3 5^2 5 3 4 2 6 5 1^2					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
5	6	0.330	3	0.267	$\circ^a$
5,3	1	0.286	4	0.086	$\circ^b$
5,3,4	2	0.106	1	0.084	$*$
5,3,4,2	1	0.065	6	0.002	$*$
1	3	0.237	5	0.173	$*$
1,3	5	0.282	6	0.001	$*$
1,3,5	6	0.057	4	0.008	—
5	6	0.319	3	0.154	$\circ$
5,3	4	0.094	1	0.028	$*$
5,3,4	2	0.111	1	0.042	$*$
5,3,4,2	1	0.058	6	0.003	$*$
6	5	0.147	1	0.079	$*$
6,5	1	0.215	3	0.013	$*$
6,5,1	3	0.073	4	0.001	—

[a] $M_{6,5}$ dominates

[b] $M_{1,3,5}$ dominates

Table 5.28:

Melody 12b: 1 2 3 7 1 2 3 7 3 4 5^2 3 4 5^2 5 6 3 1 5 6 3 1 6 5 1^2 6 5 1^2					
Working Memory	First Expectation		Second Expectation		Code
	Note	Strength	Note	Strength	
1	2	0.315	6	0.237	$*$
1 2	3	0.127	7	0.021	$*$
1 2 3	7	0.066	5	0.005	$*$
1 2 3 7	5	0.002	6	0.002	—
1	6	0.266	5	0.224	$\times^a$
1 2	3	0.103	7	0.015	$*$
1 2 3	7	0.058	5	0.006	$*$
1 2 3 7	5	0.002	6	0.002	—
3	4	0.263	5	0.118	$*$
3 4	5	0.589	1	0.001	$*$
3 4 5	6	0.027	1	0.002	—
3	4	0.155	5	0.095	$*$
3 4	5	0.525	1	0.001	$*$
3 4 5	6	0.031	1	0.003	—
5	6	0.783	3	0.083	$*$
5 6	1	0.299	3	0.065	$\circ^b$
5 6 3	1	0.127	4	0.067	$*$
5 6 3 1	3	0.043	4	0.034	—
5	6	0.557	3	0.159	$*$
5 6	1	0.376	3	0.057	$\circ$
5 6 3	4	0.133	1	0.114	$\circ^c$
5 6 3 1	3	0.059	4	0.046	—
6	5	0.297	1	0.100	$*$
6 5	1	0.300	3	0.060	$*$
6 5 1	3	0.037	2	0.003	—
6	5	6.87	1	0.047	$*$
6 5	1	1.124	3	0.427	$*$
6 5 1	3	0.061	2	0.003	—

[a] $M_{6,5,1}$ dominates

[b] $M_{6,5,1}$ dominates

[c] $M_{3,4,5}$ dominates

Table 5.29:

5.3.4 Errors

The relative performance of the network presented with each of the two input sets can be *approximately* assessed by comparing the frequency of each of the response codes in each case. For the first input set the network responded: 124 times with the correct expectation; 28 times with the second expectation correct; 4 times with both expectations incorrect; and on 67 occasions the correct expectation was impossible to make (code "—"). Thus, taking only those occasions when a correct expectation was possible, the network was correct on 79.5% of occasions, correct with its second expectation on 17.9%, and incorrect on 2.6%. For the second input set the network responded: 158 times with the correct expectation; 29 times with the second expectation correct; 4 times with both expectations incorrect; and on 41 occasions the correct expectation was impossible to make. Again taking only those occasions when a correct expectation was possible, the network was correct on 82.7% of occasions, correct with its second expectation on 15.2%, and incorrect on 2.1%.

Although these summary figures give some idea regarding **MASKART**'s performance for each input set, it is really necessary to study the network's response to each input to appreciate the underlying reasons for each error. Such inspection reveals that, of the 33 occasions on which the correct expectation is not elicited, the *primary* apparent reason (see section 5.3.2) was: post-reset bias on 1 occasion; lack of context (only 1 item in $\mathcal{F}_0$) on 3 occasions; four items in $\mathcal{F}_0$ on 6 occasions; interference from other learned sequences on 23 occasions. Often errors can be ascribed to a combination of these factors.

5.3.5 Presentation of the complete (altered) input set

The next experiment involved the presentation of the complete (altered) input set to **MASKART**. To be specific, an input pattern set representing all 12 "melodies" (altered so as to exclude in-phrase repetitions), was presented to **MASKART** 10 times. This constitutes a total of 2340 events divided into 620 phrases (25 distinct). Table 5.30 shows the expectations generated by the resulting network in response to each of the input items alone.

With the increased richness of the input pattern set it becomes more difficult to account for the expectations generated by **MASKART**. Nonetheless, observation of the network as it proceeds to equilibrium in response to the input patterns, highlights several points.

The items "4" and "7" each generate *sustained* expectations, i.e. their presentation to the

Working Memory	Expectations						
	1	2	3	4	5	6	7
1	—	0.070	0.096	0.006	0.053	0.037	0.028
2	0.524	—	0.323	0.272	0.059	0.174	0.021
3	0.383	0.181	—	0.350	0.219	0.026	0.002
4	0.018	0.437	2.181	—	8.037	0.018	0.000
5	0.201	0.030	0.167	0.130	—	0.144	0.007
6	0.166	0.119	0.024	0.029	0.554	—	0.118
7	7.145	0.169	0.014	0.005	0.020	1.782	—

Table 5.30:

network leads, in each case, to the activation of a size-2 masking field cell — $M_{5,4}$ activates in response to item "4", $M_{1,7}$ in response to "7". The reason for this is clear, the items "7" and "4" hardly ever activate alone in working memory: item "7" never initiates a phrase and therefore never appears alone in $\mathcal{F}_0$; item "4" only appears alone in $\mathcal{F}_0$ twice during presentation of the entire input set. This inevitably reduces the chances of the corresponding size-1 cell being able to activate on presentation of each item. In the absence of such a response a size-2 cell is thus likely to activate — in each case the size-2 cell which actually activates corresponds to the pairing (containing the relevant item) which appears most often in $\mathcal{F}_0$ during presentation of the input set. Note that we can see from the expectation patterns that $M_{5,4}$ faces strong competition from $M_{3,4}$, and $M_{1,7}$ competes strongly with $M_{6,7}$, as can be anticipated.

The expectations resulting from *transient* activations in response to item "1", are consistent with the continuations found in the input set. The item "1" often initiates a phrase and is most often followed by a "3", next most often by a "2" and next most often by a "5" or a "7". We can see that "3" is indeed most expected, then "2", then "5". The next most expected note is, however, a "6", in spite of the fact that item "6" never directly follows a "1". It seems that the expectation for the "6" results from interference, probably from transient activation of the size-3 cells $M_{6,5,1}$ and $M_{6,7,1}$. This interference is a clear indication that **MASKART** is not reliable in generating "transition" probabilities. On presentation of a single item, e.g.

"1", all masking field cells which have learned to represent patterns containing that item will receive input, activate to varying degrees, and thus generate expectations. Because of the low value of the working-memory shunting parameter, w, there will not be a large differential between the input received by masking field cells that respond to phrases *initiated* by the item and those that respond to phrases containing the item in another serial position.

MASKART's unreliability in indicating transition probabilities is further illustrated by an inspection of the expectations generated by items "2","3", "5" and "6". For instance, item "2" generates decreasing expectations for "1", "3" and "4", even though the pairings (2,1) and (2,3) never initiate a phrase — the expectations appear to result from the fact that "1","3" and "4" often follow item "2", when it appears in the "middle" of a phrase. Indeed, to generalize even further, the items "1", "3" and "4" are expected because they often occur in phrases containing a "2", regardless of order of occurrence. Note also that item "6" is also somewhat expected in response to a "2" — it is the item most likely to follow a phrase-initiating "2". No clear pattern emerges to allow the unambiguous interpretation of these expectations — suffice to say that the fact that *order* information does not seem to be instrumental in their generation casts doubt on the ability of **MASKART** to function as a sequential memory capable of dealing with this richer input environment.

The network was then exposed to two-item "probe" sequences, and the expectations examined. These are shown in Table 5.31, in which the expectations generated by two-item probes are shown. In each case the validity of the most expected item is indicated with the code symbols used previously. Validity is assessed with reference to the input set by ascertaining the item most likely to follow the probe sequence if that probe sequence were to *initiate a phrase* — the performance is an indicator of the network's ability to behave as a sequential memory. Performance is not good: the "correct" expectation is elicited for only 12 out of the 18 probes. On four of the remaining occasions the second most expected item is the correct one; the remaining two probes give incorrect expectations. Moreover, it is clear that even some of the correct expectations are not generated by the large cells corresponding to the whole phrase of which the probes form part. For example, the expectation of a "6" in response to the probe $(1,7)$ has more to do with transient activation of $M_{6,7,1}$ than with $M_{1,7,6,5}$. In the light of this, some of the correct responses must be considered to be "lucky guesses" and not the basis for a robust model of veridical sequential memory. The performance in response to

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1,2	0.000	0.000	0.660	0.071	0.103	0.034	0.037	$*$
1,3	0.029	0.468	0.034	0.100	0.294	0.027	0.023	$\circ$
1,5	0.000	0.090	0.085	0.038	0.000	0.097	0.041	$*$
1,7	0.000	0.111	0.117	0.002	0.061	0.650	0.000	$*$
2,4	0.487	0.000	0.985	0.000	0.597	0.126	0.004	$\circ$
2,5	0.778	0.000	0.236	0.457	0.000	0.242	0.072	$*$
2,6	0.060	0.018	0.045	0.019	0.207	0.139	0.156	$\circ$
3,2	1.578	0.000	0.002	0.595	0.040	0.056	0.014	$*$
3,4	0.058	0.284	0.000	0.000	0.707	0.005	0.000	$*$
3,5	7.084	0.053	0.000	1.076	0.000	0.253	0.002	$\circ$
4,2	0.128	0.034	0.744	0.000	0.095	0.034	0.002	$*$
4,3	0.157	0.392	0.000	0.000	0.178	0.013	0.000	$*$
5,3	0.217	0.077	0.000	5.750	0.000	0.406	0.002	$*$
5,4	0.036	0.090	0.931	0.000	0.000	0.067	0.001	$*$
5,6	0.430	0.036	0.093	0.096	0.000	0.029	0.051	$\times$
6,2	0.165	0.000	0.113	0.079	0.192	0.000	0.322	$\times$
6,5	1.391	0.057	0.259	0.245	0.000	0.000	0.096	$*$
6,7	0.766	0.359	0.007	0.002	0.066	0.000	0.000	$*$

Table 5.31:

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1,2,3	0.000	0.000	0.000	0.145	0.116	0.018	0.037	×
1,2,5	0.000	0.000	0.546	0.082	0.000	0.192	0.061	×
1,3,2	0.000	0.000	0.000	0.151	0.122	0.018	0.038	∗
1,5,3	0.000	0.324	0.000	0.215	0.000	0.267	0.039	∘
1,5,6	0.000	0.140	0.653	0.069	0.000	0.000	0.461	∘
1,7,6	0.000	0.110	0.104	0.002	0.456	0.000	0.000	∗
2,4,5	0.167	0.000	0.912	0.000	0.000	0.063	0.011	∘
2,6,7	0.201	0.000	0.021	0.013	0.146	0.000	0.000	∘
3,4,2	0.546	0.000	0.000	0.000	0.623	0.012	0.003	∘
3,5,4	0.088	0.138	0.000	0.000	0.000	0.069	0.000	×
4,2,3	0.788	0.000	0.000	0.000	1.155	0.018	0.003	∘
4,3,2	0.440	0.000	0.000	0.000	0.086	0.019	0.004	∗
5,3,4	0.035	0.024	0.000	0.000	0.000	0.016	0.000	∘
5,4,3	0.167	0.100	0.000	0.000	0.000	0.049	0.000	∘
5,4,6	0.227	0.040	0.606	0.000	0.000	0.000	0.013	∘
5,6,3	0.623	0.026	0.000	0.358	0.000	0.000	0.015	∗
6,2,3	0.980	0.000	0.000	0.254	0.230	0.000	0.070	∗
6,5,4	0.369	0.044	0.835	0.000	0.000	0.000	0.020	∗

Table 5.32:

three-item probes, shown in Table 5.32, was poor. The increased context, far from making the expectations more specific, simply activated broader regions of the masking field thus obscuring expectations from the "targeted" phrases. Only 6 probes elicited correct responses, 9 partially correct and 3 incorrect.

5.3.6 Experiments with equilibrium weights

In the previous experiment, the network was tested after being exposed to 10 passes through the complete input set. Although this resulted in significant LTM learning, it was not sufficient for the weight vectors to reach their asymptotic values. In addition, the difference in frequency of occurrence of the various phrases led to different degrees of learning for each. It was observed that this learning differential affected greatly the pattern of expectations elicited above — in particular, three-item phrases, which enjoy enhanced learning due to the presence of a "long" end note, tend to overwhelm expectations elicited from the activation of cells corresponding to less frequent four-item phrases. To address this problem, the bottom-up and top-down weights were adjusted by hand to simulate the position that would ideally be reached after the input set had been presented sufficiently often that all its constituent phrases (and sub-phrases) have been fully learned. Thus all 1,2,3 and 4-item (sub)phrases present in STM at some time during presentation of the input set were hard-wired with asymptotic weight vectors. This hard-wiring simulates "ideal" learning with reference to the current network and the behaviour of the network set up in this way illustrates fundamental weaknesses of the current approach vis-a-vis the generation of expectations corresponding to phrase continuations.

The expectations generated by the ideal network on presentation of two and three-item probes are shown in Tables 5.33 and 5.34. For the two-item probes the performance is similar to that elicited previously. 11 probes elicited the correct expectation, 5 partially correct and 2 incorrect. For the three-item probes, the performance is even worse than before: only 3 probes elicit the correct phrase completion, 11 partially correct and 4 incorrect. These results illustrate that the failure of **MASKART** to generate expectations of phrase completions is not simply a failure to learn to represent the phrases found in the input set, but a failure of the network's subsequent responses. The fact that, in the current model, it is not only *complete phrases* that become represented by masking field cells, but also certain subphrases, leads to a broad masking field response that obscures correct expectations.

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1,2	0.000	0.000	0.646	0.235	0.484	0.194	0.150	$*$
1,3	0.000	0.606	0.000	0.248	0.523	0.096	0.063	$\circ$
1,5	0.000	0.404	0.490	0.164	0.000	0.515	0.112	$*$
1,7	0.000	0.109	0.095	0.013	0.123	0.235	0.000	$*$
2,4	0.214	0.000	0.697	0.000	0.359	0.125	0.026	$\circ$
2,5	0.617	0.000	0.310	0.499	0.000	0.335	0.091	$*$
2,6	0.225	0.000	0.465	0.182	0.323	0.000	0.350	$\circ$
3,2	0.587	0.000	0.000	0.549	0.252	0.331	0.049	$*$
3,4	0.140	0.571	0.000	0.000	0.521	0.048	0.000	$\circ$
3,5	0.337	0.154	0.000	0.623	0.000	0.358	0.002	$*$
4,2	0.182	0.000	0.545	0.000	0.427	0.165	0.020	$*$
4,3	0.140	0.548	0.000	0.000	0.624	0.057	0.000	$\circ$
5,3	0.340	0.177	0.000	0.622	0.000	0.290	0.003	$*$
5,4	0.163	0.342	0.650	0.000	0.000	0.454	0.004	$*$
5,6	0.458	0.160	0.387	0.403	0.000	0.000	0.112	$\times$
6,2	0.172	0.000	0.346	0.180	0.310	0.000	0.333	$*$
6,5	0.411	0.149	0.574	0.430	0.000	0.000	0.108	$\times$
6,7	0.929	0.554	0.068	0.056	0.348	0.000	0.000	$*$

Table 5.33:

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1,2,3	0.000	0.000	0.000	0.299	0.320	0.148	0.121	×
1,2,5	0.000	0.000	0.404	0.241	0.000	0.262	0.136	×
1,3,2	0.000	0.000	0.000	0.273	0.310	0.108	0.109	∘
1,5,3	0.000	0.359	0.000	0.229	0.000	0.321	0.059	∘
1,5,6	0.000	0.311	0.377	0.166	0.000	0.000	0.238	×
1,7,6	0.000	0.262	0.195	0.019	0.411	0.000	0.000	∗
2,4,5	0.315	0.000	0.457	0.000	0.000	0.253	0.037	∘
2,6,7	0.242	0.000	0.226	0.274	0.338	0.000	0.000	∗
3,4,2	0.271	0.000	0.000	0.000	0.315	0.079	0.008	∘
3,5,4	0.116	0.251	0.000	0.000	0.000	0.208	0.000	∘
4,2,3	0.243	0.000	0.000	0.000	0.269	0.091	0.010	∘
4,3,2	0.332	0.000	0.000	0.000	0.439	0.099	0.009	∘
5,3,4	0.118	0.227	0.000	0.000	0.000	0.248	0.000	∘
5,4,3	0.120	0.249	0.000	0.000	0.000	0.275	0.001	∘
5,4,6	0.242	0.221	0.464	0.000	0.000	0.000	0.046	∘
5,6,3	0.365	0.188	0.000	0.461	0.000	0.000	0.080	∘
6,2,3	0.233	0.000	0.000	0.193	0.417	0.000	0.250	×
6,5,4	0.235	0.183	0.265	0.000	0.000	0.000	0.121	∗

Table 5.34:

A final experiment, once again using the ideally "trained" network, tested this interpretation. The three-item probes were once again presented to the network, but this time only expectations resulting from the (transient) activation of size-4 cells were registered. In other words only cells corresponding to complete phrases contributed top-down expectation. It is emphasized that, using the present network, there is no "realistic" way of imposing this constraint — the idea, however, foreshadows work presented in the next chapter. The results are shown in Table 5.35. Although the expectations are numerically small, in all cases bar one the correct item is most expected. In the remaining case, the correct item is second most expected, the error resulting from activation of $M_{4,3,2,1}$ and $M_{4,2,3,1}$ in response to the probe sequence $(1, 2, 3)$ — this further illustrates an insensitivity to item ordering.

These results suggest that progress towards a sequential memory based on the masking field might be achieved by implementing several changes. Firstly, by ensuring that only complete, familiar phrases, and not their constituent sub-phrases, become represented by masking field cells. Secondly, that expectations should be gauged with reference to only those cells which so represent a complete phrase. Thirdly, that expectations should not be dependent on the precise relative frequencies of input phrases, after a certain level of "phrase familiarity" has been reached. Fourthly, that the precise ordering of the probe sequence items should be critical in determining the expectation elicited. The network discussed in the next chapter allows us to make all these modifications. The next section contains a review of (to my knowledge) the only other piece of work employing a masking field embedded in an ART network.

5.4 Review of Gjerdingen (1992)

Gjerdingen [65] describes work which has much in common with that detailed in this chapter and the previous chapter. He, too, uses a masking field as the top layer of an **ART3** network, in the modelling of the acquisition of a sense of harmonic syntax from an input set consisting of selected works of Handel. He points out that many music-theoretical accounts of harmony assume that the listener has already internalized the "rules" of harmonic syntax and emphasizes that this knowledge is very often considered to be acquired through experience with music rather than necessarily by explicit instruction.

Gjerdingen makes a number of simplifications to the input set in order to facilitate the network's task. Firstly he selects a corpus of work with "a high degree of stylistic consistency"

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1,2,3	0.000	0.000	0.000	0.014	0.003	0.003	0.010	∘
1,2,5	0.000	0.000	0.005	0.005	0.000	0.003	0.010	∗
1,3,2	0.000	0.000	0.000	0.012	0.003	0.003	0.009	∗
1,5,3	0.000	0.004	0.000	0.003	0.000	0.013	0.002	∗
1,5,6	0.000	0.002	0.009	0.004	0.000	0.000	0.010	∗
1,7,6	0.000	0.004	0.003	0.001	0.014	0.000	0.000	∗
2,4,5	0.015	0.000	0.011	0.000	0.000	0.003	0.002	∗
2,6,7	0.004	0.000	0.003	0.000	0.008	0.000	0.000	∗
3,4,2	0.013	0.000	0.000	0.000	0.004	0.001	0.000	∗
3,5,4	0.002	0.008	0.000	0.000	0.000	0.008	0.000	∗
4,2,3	0.013	0.000	0.000	0.000	0.004	0.001	0.000	∗
4,3,2	0.019	0.000	0.000	0.000	0.005	0.001	0.000	∗
5,3,4	0.002	0.009	0.000	0.000	0.000	0.007	0.000	∗
5,4,3	0.003	0.010	0.000	0.000	0.000	0.008	0.000	∗
5,4,6	0.012	0.003	0.011	0.000	0.000	0.000	0.001	∗
5,6,3	0.015	0.002	0.000	0.015	0.000	0.000	0.001	∗
6,2,3	0.008	0.000	0.000	0.002	0.004	0.000	0.003	∗
6,5,4	0.008	0.001	0.008	0.000	0.000	0.000	0.001	∗

Table 5.35:

and alters the selected sections slightly to prevent modulation. Secondly, each harmonic "event" in the resulting set is given a label representing one of 13 "basic chordal types", each denoted by a Roman numeral which signifies the tonal function of that event relative to the current key. Each event is further labelled with values representing its duration and its place in the metre.

One of the main differences between Gjerdingen's approach and that adopted here concerns the operation of the working memory, ($\mathcal{F}_0$). Gjerdingen employs a dynamic short-term memory in which the cell corresponding to the item most recently presented is typically the most active cell, its activation subsequently decaying as cells corresponding to newly presented items compete with it. Thus the sequence of items $(1, 2, 3, 4)$ is represented by a spatial pattern of activation in which $x_1 < x_2 < x_3 < x_4$. This arrangement is similar to those discussed in Chapter 3. Gjerdingen arranges the dynamics of the short-term memory field so that the activity of a cell decays completely after the presentation of the fourth (or possibly fifth) subsequent item i.e. a maximum of four or five distinct items can be active in $\mathcal{F}_0$ at any given time.

This STM formulation severely constrains the operation of the network, particularly with reference to the generation of expectations. Suppose that the sequence $(1, 2, 3, 4)$ is presented sufficiently often "in isolation" that the masking field cell that comes to represent it, is able to activate in response to the sub-sequence $(1, 2)$. During learning, the top-down expectation pattern associated with this masking field cell will become parallel to the $\mathcal{F}_0$ pattern that excites it. If the working memory field evolves as Gjerdingen suggests then the top-down expectation pattern will be such that $E_4 > E_3 > E_2 > E_1$, where E_1 represents the magnitude of the top-down weight from the active masking field cell to the cell representing the item "1" etc. Thus when the cell representing the sequence $(1, 2, 3, 4)$ activates in response to the sequence $(1, 2)$ the expectation for the item "4" will be stronger than that for item "3", even though previous experience suggests that the reverse should be true. A similar problem occurs if an individual "performs" the item sequence represented by a given $\mathcal{F}_2$ cell out of STM (see Section 4.2). Furthermore, suppose that the sequences $(1, 2)$ and $(3, 1)$ are presented to the network with equal frequency and for equal durations so that the masking field cells A and B respectively come to represent these sequences. If the bottom-up weight from the short-term memory field cell corresponding to item "1" to the masking field cell A is denoted $z_{1,A}$ then,

for Gjerdingen's formulation, $z_{1,A} < z_{1,B}$. This being the case, on presentation of the item "1", cell B will receive more bottom-up input than cell A and will thus tend to be more successful in the competition for activation at that level. This in turn will lead to the generation of a stronger expectation for the item "3", than for the more appropriate item "2".

These problems are largely solved by the adoption of a working-memory field in which the most recently occurring items are represented by the cells with the lowest activations. The problem concerning the differential learning in bottom-up weights is still not eliminated, however, since a confusion will still result in cases where the sequence (3,1) is significantly more better than the sequence (1,2).

One advantage of Gjerdingen's formulation is that there is no need for reset of working memory, since overload is avoided. Similarly the problem of repeated items is partially solved since immediately on repetition of a given item, its corresponding cell becomes most active, indicating, correctly, that that item occurred most recently. Unfortunately the solution is only partial since all order information relating to the item's first occurrence is lost. For instance, suppose the input sequence is $(3, 1, 5, 1)$, then on the second occurrence of the item "1" there will be no indication that the item "1" occurred between the items "3" and "5" even though the cells corresponding to each of these items will remain active. Gjerdingen's formulation also fails to solve the "banana problem", in that the immediate repetition of an item is impossible to differentiate from a single (perhaps prolonged) occurrence.

Unlike **MASKART**, Gjerdingen's network allows durational and metrical factors to affect directly the pattern of activation held in the working memory. Metrical and durational prominence both result in a slightly enhanced activity for the cells representing relevant items.

The masking field in Gjerdingen's network contains 448 cells of sizes ranging from 2 to 4—there are no size-1 cells. For full coverage of all possible time orderings of all possible sets of up to 4 items chosen from 13 items, a total of over 19,000 $\mathcal{F}_2$ cells would be required (see equation 4.13). Gjerdingen avoids such an impractically large number by constructing the masking field from a set of appropriately sized and connected cells corresponding to those sequences of length 2,3 or 4 that *actually occur* in the input set. A similar, but less specific, measure was taken in **MASKART**, in which all possible sets of items were catered for, but not all time orderings of those items.

The regime under which Gjerdingen's network operates appears to differ considerably from

that described in this chapter. In contrast with **MASKART** and Cohen and Grossberg's original simulations, Gjerdingen does not allow masking-field cells to inhibit themselves (see Section 4.6)—this forces the adoption of parameter values which differ considerably from those used here and which provide for a more distributed pattern of masking-field activation. This distributed pattern, however, has the disadvantage that the level of vigilance must be set low enough to prevent frequent transmitter resets, this in turn leading to possible insensitivity to temporal ordering.

Gjerdingen only gives a small number of examples of the way in which the masking field is able to learn to recognize prototypical chord sequences contained within the input set and so it is difficult to assess the overall success or otherwise of this process. Nevertheless, he establishes that the masking field is able, at least in some cases, to exhibit behaviour that suggests that it has internalized information regarding basic harmonic syntax. In common with Section 5.1, Gjerdingen considers the transient activation of a masking-field cell whose full pattern is not yet present in working memory, as constituting a harmonic implication or expectation, although he does not detail the exact way in which such an implication affects processing at the lower level. He emphasizes that this expectation is derived not from a calculation of transition probabilities but by reference to "a repertory of learned harmonic patterns". This definition of harmonic implication is clearly analogous to the concept of melodic expectation developed here.

In summary, Gjerdingen's work closely parallels the work described in this chapter. To my knowledge no other work has been published in which a masking field, embedded in an ART network, has been used in the processing of data relating to "real world" sequential stimuli. The fact that both studies relate to the field of music perception confirms the appropriateness of the masking field model to that field, in addition to the field of linguistic perception for which it was originally developed. There are important differences between Gjerdingen's work and the work detailed here, most notably those concerned with the working-memory representation of sequential order, the setting of the vigilance parameter and the precise dynamics of the masking field. Some of these differences can be attributed to the difference in application of the two studies—it's possible that a more distributed pattern of masking field activation is better suited to the processing of patterns relating to harmonic progressions. Gjerdingen's is an important contribution to the study of self-organizing networks in general and of their

application to the modelling of music perception in particular.

5.5 Summary

5.5.1 Problems with the current formulation

The incorporation of a Cohen-Grossberg masking field within an ART network is a task fraught with problems: some of these have been successfully addressed in this chapter; others have proved intractable. The intractability of these problems results from a fundamental incompatibility between various design constraints imposed during the network's evolution. In this section I will describe some of the problems encountered during the development of **MASKART**, highlighting those which *fundamentally* limit such a network. The discussion of these problems will be divided, somewhat artificially, into sections describing the problems of each "component" of **MASKART** (the masking field, the **ART3** network and the working memory) and finally the problems encountered in trying to unite these components into the single **MASKART** network.

Problems with the masking field

- One of the most important problems faced in the modelling of a masking field is the combinatorial explosion in the number of $\mathcal{F}_2$ cells required as the size of $\mathcal{F}_0$ increases. In their simulations Cohen and Grossberg use a working memory capable of representing 5 different items, for which the number of $\mathcal{F}_2$ cells needed for "full coverage" of all sequences of 3 items or fewer, is 85 (see Equation 4.13). They actually employ a masking field containing 100 cells, divided into only three sections (1,2 and 3), which still falls short of providing full coverage, the number of cells per population being set at 4 for all $\mathcal{F}_2$ populations (note that this is not sufficient to cover all possible time orderings of a set of three items). For the modelling of the perception of even very simple (and simplified) melodies, a minimum of seven $\mathcal{F}_0$ cells and four $\mathcal{F}_2$ sections are required. For full coverage, these figures suggest a masking field containing 1099 cells—a number too large for practical simulation. **MASKART** employed 336 $\mathcal{F}_2$ cells, the reduction being achieved by emulating Cohen and Grossberg in limiting the number of cells per population. This measure limits the ability of the masking field to cope with certain

input environments. Gjerdingen [65] chose to limit the number of $\mathcal{F}_2$ cells in a different way (see Section 5.4).

Nigrin [117] points out the consequences of building a *hierarchy* of masking fields each offering full coverage: assuming $\mathcal{F}_0$ has 20 items and $\mathcal{F}_2$ has four sections, $\mathcal{F}_2$ would require 123,520 cells. A higher-level masking field, having $\mathcal{F}_2$ as its input field, would require approximately 2×10^{20} cells—roughly the total number of human brain cells currently in existence! Clearly Cohen and Grossberg's model is severely limited in this respect.

- The hard-wiring of the masking field also causes another problem, identified earlier with respect to the generation of expectations. Since all item combinations of up to a given length are initially "wired-up", and the *a priori* behaviour of the network is to respond most strongly with a masking-field cell appropriate to the length of the sequence represented in STM, the masking field learns to represent all sequences "leading up" to a given phrase, rather than just the phrase itself. As we have seen, it takes specific conditions on the timing of inputs to bring about a situation in which the sustained activation of a larger cell can respond to a short sequence (sustained expectation) — such a situation is then irreversible. This profusion of "representative" cells is undesirable.

- The setting of parameters for masking fields of different sizes is also problematic. The approach adopted here was to leave unchanged as many as possible of the (corrected) parameter choices used in Cohen and Grossberg's simulations, using only variations in the parameters C and F in order to attempt to elicit the correct behaviour from $\mathcal{F}_2$. This approach was moderately successful, though it was discovered that the parameter space had to be specifically investigated for each different network used. In addition, it proved difficult to model large masking fields for which a "distributed" pattern of activation was required at $\mathcal{F}_2$.

- The working memory used by Cohen and Grossberg does not obey the "LTM Invariance Principle". When this principle is enforced, the shunting parameter, w, must be set close to unity to ensure correct masking field behaviour. This causes problems when the masking field is embedded in an ART network (see later).

- The learning rule used by Cohen and Grossberg is deficient for the reasons given in Section 4.9.4.

Problems with the working memory

- The model of working memory developed in [68] is limited by the fact that it cannot represent repeated items. This has been referred to as the "banana problem". As a result **MASKART** cannot learn common melodic phrases such as $(1, 2, 3, 1)$, and the rules for $\mathcal{F}_0$ reset must be adjusted in a rather *ad hoc* fashion.

- The version of the working memory used in **MASKART** uses a constant shunting parameter, w, and is static, in the sense that its operation is not modelled with a differential equation describing the changes in working memory activation with time. Such a working memory does not exhibit the properties described in Section 4.2, which would otherwise enable it to account for certain serial recall effects.

- The adoption of a constant shunting parameter, w, does not allow the rhythmic and metrical aspects of a given sequence presentation to be reflected in the resulting $\mathcal{F}_0$ spatial pattern. Metrical factors were, however, reflected in the rules for $\mathcal{F}_0$ resets due to "overload" and item repetition.

Problems with ART3

- **ART3** dynamics can lead to the repeated firing of inappropriate $\mathcal{F}_2$ cells (see Section 4.9.2). In **MASKART**, $\mathcal{F}_2$ cells that are responsible for mismatch reset are prevented from immediately reactivating by the temporary suppression of their ability to emit inhibitory signals—in the context of **ART3** this can be thought of as being due to transmitter depletion at inhibitory synapses. This measure is similar to that used in **ART2**, in which a similar result is obtained by temporarily clamping the activation of such cells to zero.

- Modelling **ART3** transmitter dynamics is computationally expensive and, in the light of the problem discussed above, may not be of sufficient practical benefit to justify the use of the **ART3** algorithm (as opposed to **ART2)** in large networks.

Problems with MASKART

- A critical problem with **MASKART**, and indeed any network which embeds a masking field within an adaptive resonance loop, concerns the specification of the conditions for transmitter reset. In **MASKART** one of the fundamental constraints was that the network should be able to distinguish between different orderings of the same set of items. The use of a very high vigilance parameter was therefore necessitated by the low value of the shunting parameter, w. The need for this high vigilance parameter restricts the operation of the masking field considerably. For instance, it makes the *sustained* coactivation of two or more $\mathcal{F}_2$ cells, in response to a given input, extremely unlikely (see Section 5.2.2). This result in turn contradicts one of the basic tenets of the masking field model , namely that a distributed $\mathcal{F}_2$ spatial pattern can represent a weighted set of alternative "parsings" of the currently active working memory pattern. The solution used in **MASKART** was to allow alternative parsings, in particular those indicating expected continuations, to be represented either by the *transient* coactivation of $\mathcal{F}_2$ cells, in a period during which mismatch reset was suppressed, or by the *sustained* activation of a single $\mathcal{F}_2$ cell representing a superset of the simultaneously active working memory pattern. The latter situation could only result from an input history which strongly "justified" such a sustained expectation—the justification must be strong since **MASKART** suffers from the disadvantage that the ability of a given (small) pattern to activate an $\mathcal{F}_2$ cell representing a larger pattern cannot be reversed, i.e. the smaller pattern will never subsequently come to be represented itself at $\mathcal{F}_2$, no matter how often it is presented. Furthermore, **MASKART** is unable to indicate that a given sequence, say $(1,2,3,4)$, should best be parsed into two sub-sequences, e.g. $(1,2)(3,4)$, firstly because, as noted previously, the resultant coactivation of the two corresponding $\mathcal{F}_2$ cells would almost certainly lead to a mismatch reset, and secondly because there will inevitably be a bias against reactivating the cell representing the pattern $(1,2)$, since it will have been recently reset on presentation of the item "3".

  The problems listed in the previous paragraph are the inescapable result of embedding a masking field within a high-vigilance ART network. Since the possibility of operating under a lower-vigilance regime is excluded by the necessity to distinguish between different time orderings of the same set of items, these problems constitute serious lim-

itations on the operation of **MASKART**, or indeed any other network combining the Cohen-Grossberg masking field with an ART architecture.

- Even within the limitations described above, **MASKART** has some difficulty in reliably learning to distinguish different orderings of the same items. For instance, suppose that the patterns $(1,2,3)$ and $(1,3,2)$ are each repeatedly presented at approximately the same frequency. Under slow-learning conditions it is possible for an $\mathcal{F}_2$ cell to learn a pattern which is the *average* of the two individual patterns. This $\mathcal{F}_2$ cell may be able to activate and "pass the vigilance test" on presentation of *either* pattern and thus that $\mathcal{F}_2$ cell may come to represent *both*.

- A more general problem with **MASKART** is that it does not readily lend itself to being incorporated within a hierarchy of similar networks, not least because of the problem with the combinatorial explosion in required cell numbers (discussed above). Such a hierarchy would be enable the learning of "higher-order" regularities in a given input environment, for example the fact that, say, the sequence $(4,5,6)$ often follows the sequence $(1,2,3)$.

In the next chapter I shall introduce and apply the work of Nigrin [117][118], who has developed a network (SONNET) that retains the desirable properties of a masking field embedded in an ART architecture, whilst solving many of the problems listed above.

Chapter 6

SONNET 1

6.1 Introduction

In the previous chapters, the Cohen-Grossberg masking field was introduced and details were given of how this network was incorporated within an adaptive-resonance architecture. The resultant network, **MASKART**, was applied to the task of modelling the perception of melodic sequences, with particular emphasis being placed on the generation of melodic expectations. Although **MASKART** proved inadequate in several respects, the simulations in which it was used were instructive in suggesting a number of possible improvements. This chapter will introduce **SONNET 1**, a network recently developed, with such improvements in mind, by Nigrin ([117] [118]) — I shall summarize the ways in which it differs from **MASKART** and how these differences solve many of **MASKART**'s problems. The results of simulations of **SONNET 1**, using the same input-set as in Chapter 5, will be described, as will modifications to the network introduced in the course of these simulations.

6.2 SONNET 1

This section will give details of the operation of the **SONNET 1** network. The description given here is based on that presented in [117] and [118], but necessarily omits much detail—it will be chiefly directed towards describing the specific measures Nigrin has taken to address problems similar to those encountered with **MASKART**. A **SONNET 1** network consists of two fields: an input field (which can be configured to behave as a working memory) and a

classification field—these will be referred to as $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively.

6.2.1 SONNET 1 is initially homogeneous

Unusually amongst neural network models, the masking field is non-homogeneous: masking field cells are initially connected only to a subset of the cells in the input field; they have a size which varies with the magnitude of their receptive field, and a set of parameters in turn related to their size; the strengths of the inter-field inhibitory connections also vary. We have seen how the "hard-wiring" of this non-homogeneity forces a combinatorial explosion in the size of any masking field that is to be able to cope with unpredictable input environments. It was shown that this constituted a serious problem for the applicability of the masking-field model, particularly in cases where the number of different items able to be represented in the working memory becomes large. Nigrin addresses this problem by ensuring that **SONNET 1** is initially homogeneous and that subsequent differentiation is introduced only in response to the network's input environment. This process introduces a good deal of complexity into the network, since not only the bottom-up and top-down LTM traces but also $\mathcal{F}_2$'s lateral inhibitory weights and cell sizes must be learned. The precise way in which the learning of these values is achieved will be described later. First it is necessary to introduce a crucial aspect of **SONNET 1**'s operation.

6.2.2 The calculation of $\mathcal{F}_2$ input

In the previous chapters it was noted that, for values of the shunting parameter significantly greater than unity, the magnitude of the weighted input to $\mathcal{F}_2$ was insensitive to changes in the activation of $\mathcal{F}_1$ cells responding to items near the end of a list — this necessitated the adoption of a low shunting-parameter (w), which in turn led to difficulties in discriminating patterns corresponding to sequences consisting of different orderings of the same items. Nigrin's approach is to take the unusual step of introducing non-linearity into the calculation of $\mathcal{F}_2$ input. Thus in **SONNET 1**

$$b_i = I_i^+ I_i^\times \tag{6.1}$$

where

$$I_i^+ = \sum_{j \in \mathcal{F}_1} S_{ji} z_{ji}^+ \tag{6.2}$$

$$I_i^\times = \max(\prod_{j \in T_i} I_{ji}^\times, K_8) \tag{6.3}$$

$$I_{ji}^\times = K_1 + K_2 \min(1, \frac{S_{ji}}{Z_{ji}}) \tag{6.4}$$

$$Z_{ji} = \frac{z_{ji}^+}{(\sum_{k \in T_i} (z_{ki}^+)^2)^{1/2}} \tag{6.5}$$

$$S_{ji} = \begin{cases} \frac{s_j}{(\sum_{k \in T_i} (s_k)^2)^{1/2}} & \text{if } j \in T_i \\ \frac{s_j}{(\sum_{k \in \mathcal{F}_1} (s_k)^2)^{1/2}} & \text{otherwise} \end{cases} \tag{6.6}$$

where s_i is the activity of the i_{th} $\mathcal{F}_1$ cell, z_{ji}^+ is the excitatory weight from the j^{th} $\mathcal{F}_1$ cell to the i^{th} $\mathcal{F}_2$ cell, T_i is the set of indices corresponding to the set of $\mathcal{F}_1$ cells to which the i^{th} $\mathcal{F}_2$ cell is "significantly" connected (see later) and b_i represents the bottom-up input to the i^{th} $\mathcal{F}_2$ cell assembly, χ_i. The cell assembly, χ_i is made up of the set of cells b_i, c_i, d_i and e_i, the same symbol being used both for the name and for the activation of a given cell. A diagram of a cell assembly is shown in Figure 6.1. All parameters of the form K_n are constants. The term I_i^+ is the usual weighted activation term, i.e. the inner product of the *normalized* input and the weight vector. The term $I_i^\times$ compares how well the current normalized input pattern matches the cell's weight vector (normalized over the set T_i)—Nigrin refers to this as the "confidence" of the $\mathcal{F}_2$ cell. The use of $I^\times$ ensures that even small changes in the input pattern can change significantly the bottom-up input to a given cell. In particular, the pattern of bottom-up input to a cell can change significantly depending on the presence or absence of items at the end of a given list. This modification permits the use of a shunting parameter significantly greater than unity. Nigrin shows how *two* values can both be communicated by a single pulse-train signal and thereafter assumes that each $\mathcal{F}_2$ cell outputs a signal conveying both its own activation and its confidence value—the following sections will show how each is employed.

6.2.3 The learning of bottom-up LTM traces

The bottom-up LTM traces are modified using the following differential equation

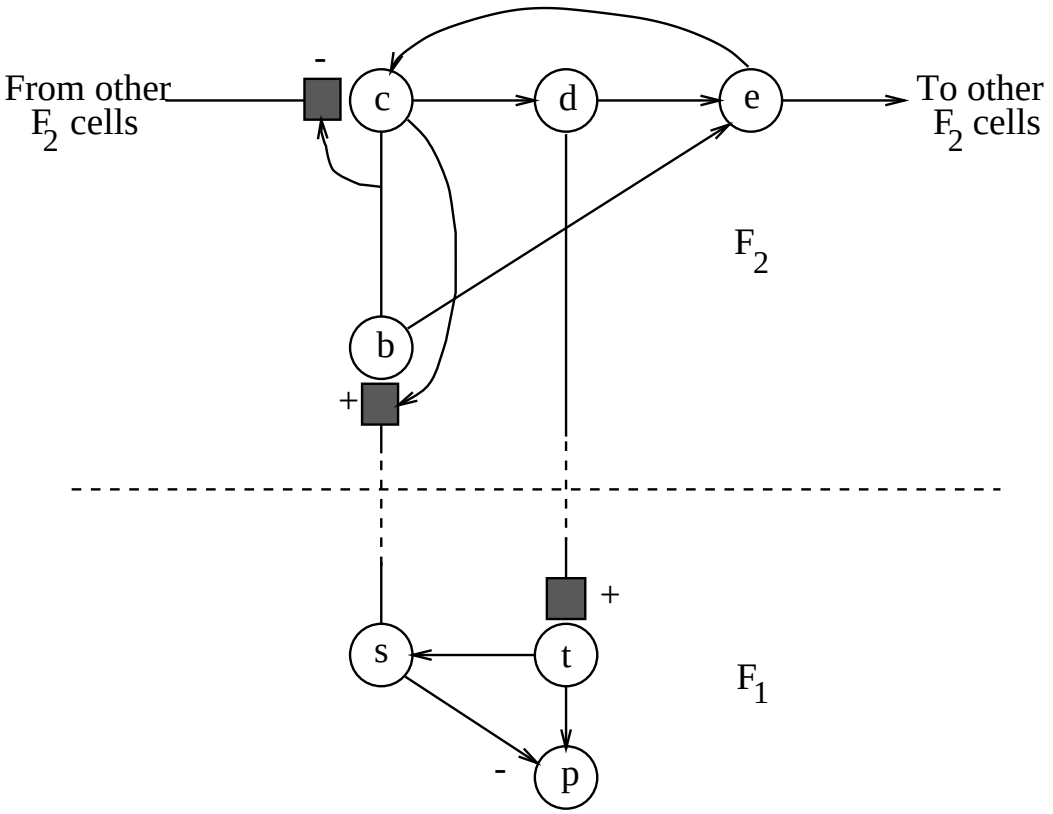


Figure 6.1: The arrangement of cells in $\mathcal{F}_1$ and $\mathcal{F}_2$

$$\frac{d}{dt}z_{ji}^{+} = \epsilon_1\epsilon_2 r_{ji}c_i[-L_i z_{ji}^{+} + S_{ji}c_i] \tag{6.7}$$

where $\epsilon_1\epsilon_2$ is the learning rate. This is similar to the learning equation used in **MASKART**, in that the bottom-up weight vector becomes parallel to the input vector at a rate dependent on the activity of the $\mathcal{F}_2$ cell. It differs in that the term c_i also multiplies the input term, causing "the maximum value of the weights to depend on the value of c_i". This is important in ensuring that a given input pattern will not be learned by an $\mathcal{F}_2$ cell if there already exists an $\mathcal{F}_2$ cell which has learned a sufficiently similar input pattern — "sufficient similarity" is measured, as in ART networks, relative to a vigilance parameter. In short, a cell which has previously learned a given input pattern will inhibit the activation of other cells on presentation of that input pattern (and similar patterns)—this keeps the activations, c, of such cells low and thus prevents them from fully learning the input pattern. The mechanism is discussed in more detail later. The term L_i is set to zero at $\mathcal{F}_2$ cells whose bottom-up weight vector is small (this ensures that the size of all bottom-up weight vectors stays above some minimum value) and is set to a constant, L, otherwise. In addition, when sufficient learning has taken

place at a given $\mathcal{F}_2$ cell, such that $I_i^+ \geq 1.0$, all learning at that cell is stopped to ensure stability. A cell assembly at which the weights are thus "frozen" will be referred to as being "committed".

Finally, the term r_{ji} regulates the learning process in quite a subtle way. To compute r_{ji}, Nigrin introduces two further parameters, M_i and I_i^{max}. The variable I_i^{max} represents the highest value of $I^\times$ previously encountered by the i^{th} $\mathcal{F}_2$ cell. M_i represents the degree to which the *current* spatial pattern at $\mathcal{F}_1$ is matched by the LTM weights to the i^{th} $\mathcal{F}_2$ cell, and is given by

$$M_i = \min(1, \frac{I_i^\times}{I_i^{max}}) \tag{6.8}$$

Letting K_5 and K_6 be constants and z_i^{tot} equal the length (L_2-norm) of the i^{th} bottom-up weight vector, r_{ji} is given by

$$r_{ji} = \begin{cases} 1 & \text{if } z_i^{tot} < K_5 + K_6 \\ (\frac{z_i^{tot} - K_5}{K_6})^{1-M_i^{-4}} & \text{if } j \in T_i \text{ or } \frac{d}{dt} z_{ji} < 0 \\ (\frac{z_i^{tot} - K_5}{K_6})^{1-(1-M_i)^{-4}} & \text{otherwise} \end{cases} \tag{6.9}$$

In this way the value of r_{ji} is allowed to affect learning differently depending on whether j is a member of the set T_i. Basically, if the weight vector to the i^{th} $\mathcal{F}_2$ cell matches well the current $\mathcal{F}_1$ pattern (high M), then the "significant" weights are allowed to change rapidly, whilst the insignificant weights are prevented from doing so; if there is a bad match (low M) then the reverse is true. Note that when $z_i^{tot} < K_5 + K_6$, r_{ji} is set to one and thus does not affect bottom-up weight learning.

The provision of the term r_{ji} allows the network to learn quickly so-called "embedded" patterns. For instance, the pattern $(1, 2)$ can be learned from the sequences $(1, 2, 5, 6), (4, 3, 1, 2)$ and $(5, 1, 2, 6)$. It also allows the patterns $(1, 2)$ and $(2, 1)$ to be learned separately, rather than the average of the two patterns being learned, even if they are presented in alternation—this was a problem with **MASKART** (see Section 5.5.1).

6.2.4 Specification of the set T_i

In Section 6.2.2, the set T_i was described as the set of indices corresponding to the set of $\mathcal{F}_1$ cells to which the i^{th} $\mathcal{F}_2$ cell is "significantly" connected. To be more accurate, the set

T_i is the set of indices corresponding to the set of $\mathcal{F}_1$ cells which tend to have a significant activation at times when c_i is also high. Since, by the learning rule, one would expect strong bottom-up LTM weights to form between cells whose activity is correlated in this way, the two descriptions are *almost* interchangeable. The set T_i is specified by

$$j \in T_i \text{iff } K_3 w_{ji} \geq w_{0i} \tag{6.10}$$

where w_{0i} is the largest weight, w_{ji}, to χ_i. The weights, w_{ji} are given by

$$\frac{d}{dt} w_{ji} = \begin{cases} \epsilon_1 \epsilon_2 r_{ji} c_i [-L_i w_{ji} + c_i] & \text{if } K_4 S_{ji} > S_{0i} \\ \epsilon_1 \epsilon_2 r_{ji} c_i [-L_i w_{ji}] & \text{otherwise} \end{cases} \tag{6.11}$$

It should be noted that the degree of learning of w_{ji} is not dependent on the activity of the j^{th} $\mathcal{F}_1$ cell, other than in the determination of which cells have *significant* activation.

The set T_i must be specified so that the calculation of $I_i^{\times}$ only depends on the activity of those $\mathcal{F}_1$ cells which are active in χ_i's classified or near-classified pattern. In addition, information pertaining to the set T_i is used in the calculation of S_{ji}, Z_{ji} and r_{ji}. The evolution of the set T_i shows how the bottom-up connectivity pattern characteristic of a masking field, whereby each $\mathcal{F}_2$ cell is connected only to a subset of the $\mathcal{F}_1$ cells, can emerge from a network that is initially "fully connected."

6.2.5 The evolution of $\mathcal{F}_2$ activations

The activity of the cell c_i at $\mathcal{F}_2$ is given by

$$\frac{d}{dt} c_i = \epsilon_1 \left[-A c_i + \frac{B - c_i}{D_i} (v_2 b_i + e_i) - \frac{c_i}{D_i} v_3 Q_i \right] \tag{6.12}$$

where A is a decay constant, B is the cell's maximum activity and v_2 and v_3 are constants. The parameter ϵ_1 allows the time-axis to be "scaled". Q_i represents the total lateral inhibitory input to the i^{th} $\mathcal{F}_2$ cell and is given by

$$Q_i = \begin{cases} \sum_{j \neq i} e_j z_{ji}^- & \text{if } \chi_i \text{ is committed or } (\sum_{j \neq i} e_j z_{ji}^- < K_7 e_i^{big}) \\ K_7 e_i^{big} & \text{otherwise} \end{cases} \tag{6.13}$$

where e_i^{big} is the largest current signal from any uncommitted e cell. Note that the total inhibition to an uncommitted cell is limited. This allows uncommitted cells to *begin* to learn

patterns even if the patterns, or patterns similar to them, are already classified—as noted earlier, this learning process can later be suppressed to prevent multiple $\mathcal{F}_2$ cells from representing the same input pattern. To limit the number of uncommitted cells that can at least *begin* learning, Nigrin applies the following rule:

> "If there exist q uncommitted assemblies, χ_j, such that $K_{10}e_j > e_i$ and q uncommitted assemblies, χ_k, such that $I_k^\times > I_i^\times$ (the q cells may be the same or different) then c_i is set to zero".

The value D_i is called the dilution parameter and it corresponds to the "size" of the i^{th} $\mathcal{F}_2$ cell, as defined with reference to the masking field. Changes in D_i are given by

$$\frac{d}{dt}D_i = \begin{cases} 0 & \text{if } \chi_i \text{ is committed} \\ \epsilon_1\epsilon_2\epsilon_3 c_i(-D_i + I_i^\times) & \text{otherwise} \end{cases} \tag{6.14}$$

D_i thus "tracks" the value of $I_i^\times$.

The activations of the other $\mathcal{F}_2$ cells are given by

$$d_i = \min(c_i, 1.0) \tag{6.15}$$

$$e_i = b_i(d_i)^2 \tag{6.16}$$

6.2.6 The learning of $\mathcal{F}_2$ lateral inhibitory weights

Recall that uncommitted $\mathcal{F}_2$ cells are prevented from learning to classify patterns that are already classified, by virtue of the fact that their activity remains low due to lateral inhibition from the relevant committed cell (remember the size of the bottom-up weight vector depends upon c_i). The lateral inhibitory weight, z_{ji}^- from e_j to c_i is modified as follows

$$\frac{d}{dt}z_{ji}^- = \begin{cases} \epsilon_1\epsilon_2\epsilon_3 c_i(-z_{ji}^- + 1) & \text{if } (\rho_i \le I_j^\times) \text{ or both cells are committed} \\ \epsilon_1\epsilon_2\epsilon_3 c_i(-z_{ji}^- - 1) & \text{if } (\rho_i > I_j^\times) \text{ and } (z_{ji}^- > 0) \\ 0 & \text{otherwise} \end{cases} \tag{6.17}$$

where ϵ_3 is typically greater than 1, allowing the lateral inhibitory weights to vary faster than the bottom-up excitatory weights, and where

$$\rho_i = \begin{cases} (I_i^\times)^\rho & \text{if } I_i^\times \ge 1 \\ (I_i^\times)^{(1/\rho)} & \text{otherwise} \end{cases} \tag{6.18}$$

This allows uncommitted cells, which have begun to classify a given pattern, to become "uninhibited" by cells which have already learned similar but not identical patterns. The variable ρ is the vigilance parameter, which specifies how different the new pattern must be from previously classified patterns to allow it to be classified by an $\mathcal{F}_2$ cell of its own. As ρ approaches unity, "the closer an active pattern must match an existing category to prevent a new category from forming". This vigilance parameter is thus similar to that found in ART networks, in that it specifies the "breadth" of the categories formed. Importantly, the mechanism by which the vigilance parameter influences learning in **SONNET 1** does not make use of a reset signal—this allows **SONNET 1** to avoid many of the problems associated with reset, described in the previous chapter.

A further measure ensures that cells that classify non-overlapping patterns do not compete with each other. If cell assemblies χ_i and χ_j are both committed and there exists a time when $I_i^{\times} > 0.95 I_i^{max}$ (indicating that χ_i's pattern is almost fully present at $\mathcal{F}_1$) and $I_j^{\times} \approx K_1^{|T_j|}$ (indicating that none of χ_j's pattern is present at $\mathcal{F}_1$), then the inhibitory weight z_{ij}^{-} is set to, and frozen at, zero. Note that the condition on $I_j^{\times}$ differs slightly from that given in [118] (which is unstable) and will be further modified later in the chapter. Both conditions can only be satisfied simultaneously if the two cells classify entirely separate patterns. In this way, a "large" but unfamiliar input pattern can be decomposed into two or more smaller but familiar patterns. This behaviour was not possible with the masking field or with **MASKART**, for the reasons discussed in the previous chapters.

6.2.7 Configuring $\mathcal{F}_1$ as a dynamic working memory

Nigrin [117] describes how $\mathcal{F}_1$ can be configured as a dynamic working memory. As in the previous chapter, this working memory converts an item sequence into a spatial pattern, such that the item order is stored by the relative cell activations. Unlike the model presented in the previous chapter, the working memory in **SONNET 1** is fully dynamic, being modelled with a differential equation specifying the change in cell activation, s_i, over time

$$\frac{d}{dt}s_i = (B_1 - s_i)[v_1 s_i + I_i^{(1)}] - s_i \sum_{j \neq i} v_1 s_j \tag{6.19}$$

where B_1 is the maximum cell activation, $I_i^{(1)}$ is the cell's external input and v_1 is a constant. The parameters B_1, v_1 and $I_i^{(1)}$ can be varied, along with the time for which external

input is applied, to produce a working memory field with an approximately constant shunting parameter, w. The particular choice used in [117], which will also be adopted here, gives a working memory that has $w \approx 2$ for an input environment in which the items are presented isochronously.

The issue of working memory overload is dealt with as before. Overload is deemed to occur when the sum of the activity at $\mathcal{F}_1$ exceeds a certain value, K_{13}, at which point a reset signal is triggered, which quenches the activity of all $\mathcal{F}_1$ cells whose activity exceeds a given threshold, K_{14}. This threshold can be varied so as to allow a specified number of items to "drop out" of working memory at reset. In addition, Nigrin introduces a random element in the setting of the threshold.

In the previous chapter the detail of the reset process was altered somewhat, in order to make it more compatible with the musical task to which the network would be applied. The random element described above was abandoned and instead the reset process allowed all events previous to the most recent "accented" event to be dropped from working memory. An accented event was considered to be one which initiated a metrical and linguistic unit. Likewise, a non-immediate event repetition also caused a reset that eliminated from working memory at least those events previous to the most recent accented event. The same approach could be used for the operation of **SONNET 1**'s working memory, $\mathcal{F}_1$, with the provisos regarding self-organization noted in the previous chapter. In fact this chapter does not deal with the presentation of extended sequences and the discussion is taken up again in Chapter 7.

The likelihood of $\mathcal{F}_1$ overload in **SONNET 1** is reduced as learning proceeds, due to the introduction of a mechanism for "chunking" classified patterns out of working memory. This requires additional network structure, specifically the provision of a top-down adaptive filter from $\mathcal{F}_2$ to $\mathcal{F}_1$. This will be described in the next section.

6.2.8 The addition of feedback weights

Nigrin [117] describes how the provision of feedback (top-down) weights allows **SONNET 1** to deal with arbitrarily long item lists. The feedback weights are initially set to zero and their subsequent variation is given by

$$\frac{d}{dt} z_{ij}^f = \epsilon_1 \epsilon_2 (d_i)^2 (-z_{ij}^f + S_j d_i) \tag{6.20}$$

where z_{ij}^f is the feedback weight from χ_i to s_j and $S_j = s_j / \sum_{k \in F_1} s_k$. This is very similar to the top-down learning equation used in the previous chapter. Thus the feedback weights from χ_i become parallel to χ_i's classified pattern. To prevent the learning of inappropriate weights after classification, the following constraints are applied. Once χ_i is committed, its feedback weights are only permitted to change if $I_i^{\times} \geq 0.95 I_i^{max}$. If $s_j = 0$ at any time whilst this condition is satisfied, then z_{ij}^f is set to, and frozen at, zero. Furthermore, if t denotes the first time at which the above condition is satisfied simultaneously with the condition $\frac{d}{dt} z_{ij}^f > 0$, then at times subsequent to t, learning is only enabled if both conditions are again satisfied—this is to prevent the possibility of inappropriate weight decay.

Lists longer than the transient memory span can now be processed by resetting $\mathcal{F}_1$ cells whose activity forms part of a pattern classified at $\mathcal{F}_2$. Classification is assumed to occur if the activity c_i of a committed $\mathcal{F}_2$ cell continuously exceeds a high threshold, K_{11}, for a time K_{12}, where K_{12} is approximately equal to the time required for the presentation of two items. The fact that c_i must maintain a high level of activity during the presentation of subsequent items, prevents the network from, for instance, erroneously classifying the pattern $(1, 2, 3)$ during presentation of the pattern $(1, 2, 3, 4)$, when both patterns have been learned, or are required to be learned, at $\mathcal{F}_2$. When classification is deemed to have occurred, the activity, c_i, of the relevant $\mathcal{F}_2$ cell is set to zero. The fact that a pattern has been classified at $\mathcal{F}_2$ is communicated to its constituent $\mathcal{F}_1$ cells by the large, sudden drop in feedback to these cells, which results from the resetting of the classifying cell. When an $\mathcal{F}_1$ cell receives such a drop in feedback, its own activity is quenched (in fact, this mechanism will be modified for reasons presented later in the chapter). This process eliminates "classified" items from working memory, allowing cells representing unclassified items to remain active. In other words, the classified pattern is "chunked out" of working memory. Note that, as a result of the conditions for classification, familiar sequences ending with an event of long duration (or succeeded by a pause) are likely to be chunked out of working memory. Issues relating to the rhythm of presented sequences will be discussed in greater detail in the following section.

6.3 Timing Issues

In Section 6.2.7 it was noted that, for a particular choice of parameters B_1, v_1 and $I_i^{(1)}$, $\mathcal{F}_1$ could be configured to behave as a working memory with a constant shunting parameter,

$w \approx 2$, when presented with a sequence of items, each pair of successive items having the same inter-onset-interval, t_{IOI}. A problem can occur, however, if the rate of presentation of items to a working memory field varies significantly from $1/t_{IOI}$, whilst the other parameters remain constant—very different values for the shunting parameter can result. This problem is particularly acute for working memory fields at different levels in a hierarchy (as in the next chapter)—a given field is likely to have an item presentation rate which differs considerably from that found at a field elsewhere in the hierarchy. To avoid having to set the parameters at each field to be consistent with the particular rate of item presentation at that field, Nigrin [117] suggests a modification that makes the network "event driven" rather than "time driven". This requires that all changes in cell activities s_j, b_i and c_i in a given $\mathcal{F}_1$-$\mathcal{F}_2$ pair, as well as all modifications of their LTM weights, are enabled only for a fixed time period after the onset of an item at $\mathcal{F}_1$. Nigrin suggests that this gating can be achieved using a non-specific attentional arousal signal, which is active for a short time after the onset of an item at $\mathcal{F}_1$. Such a constraint, however, can lead to two separate, but related, problems

- Different presentation rhythms cannot be reflected in the spatial pattern that evolves at the working memory, $\mathcal{F}_1$. This is because the time during which the spatial pattern is able to change in response to an item, is independent of either the duration of the item itself or the inter-onset-interval defined by it and the successive item.

- The effect noted in previous chapters, whereby sequences ending in events of relatively long duration enjoy enhanced LTM learning, is lost if the modification of LTM weights is disabled at times outside of the attentional "window".

Fortunately it is possible to solve both, or either, of these problems by making some assumptions about the nature of the attentional signal. We assume that the attentional signal is active not only at times immediately after an item has occurred, but also at times immediately after an item is *expected to occur*. In this case the term "expected" does not refer to a "veridical" expectation, such as might be experienced when listening to a melody whose sequence of pitches and durations is familiar, but to the "schematic" expectation for event occurrence, generated by a constant or near-constant beat period. Gjerdingen [62] discusses and simulates a "neural metronome" that concentrates attentional resources at points in time at which events are likely to occur, namely "on the beat". Chapter 8 describes simulations of

a network of cells, based on a similar network developed to model the operation of circadian rhythms in mammals, that is able to exhibit activity that varies rhythmically, in sympathy with an external rhythmic stimulus. The metronome analogy is useful: if we assume that the attentional signal is active for a short period after *each beat of a metronome* accompanying the playing of a melody, *whether or not an event occurs at that beat*, then one, or both, of the problems detailed above can be solved. If we wish different presentation rhythms to affect the spatial pattern at $\mathcal{F}_1$, then we allow changes in activity at this field to be gated by the "attentional metronome", rather than by the occurrence of actual items; if we wish the inter-onset-intervals to affect LTM learning, then we gate LTM changes likewise. Note that by choosing the way in which we allow the attentional signal to affect changes at each point in the network, we can selectively solve, as the application demands, either of the potential problems listed above.

6.4 Simulating SONNET 1

In order to apply **SONNET 1** to the processing of melodies, it was necessary to write a computer program to simulate its operation. The program was written in Objective-C on the NeXT computer and was designed to be as flexible and as easy to use as possible. In contrast with the **MASKART** program all the parameters used in the description of the network, including those specifying the size of each field, are continuously on view and are able to be modified by the user at run-time—this necessitated a considerable amount of dynamic memory-allocation. Similarly any data files, representing input patterns or previously learned LTM information, are able to be loaded by means of a "pop-up" file browser, rather than needing to be specified in the command line. Animated plots of the values of the activities s_j, c_i and the amount of top-down feedback received by the $\mathcal{F}_1$ cells, are provided on-screen and at any point the animation can be paused to give a snap-shot of the state of the network. The animated displays can be disabled if run speed is crucial. As in the previous chapter, a fourth-order Runge-Kutta algorithm is used to integrate the differential equations relating to the change in cell activations, the constant step size being set small enough to ensure stability; a first-order Euler algorithm is used in the integration of the differential equations relating to the change in LTM weights. The input files are similar to those used in the previous chapter—they include data relating to the scale degree and duration of each note event.

Once the network has been exposed to a set of input patterns, data relating to the resultant state of the network is able to be stored in a series of files. In addition to the bottom-up, top-down and lateral inhibitory LTM weights, a large amount of other information is stored. This includes the size, D_i, of each $\mathcal{F}_2$ cell, values of I_i^{max} and various "flags" indicating, amongst other things, which LTM weights are frozen and which $\mathcal{F}_2$ cells are committed.

A screen dump showing the appearance of the program's interface during operation is shown in Figure 6.2. This also shows typical parameter settings — the rationale for the setting of particular parameters is given in Section 6.5.2.

The program was used to repeat many of the simulations described in [117][118] —this not only provided an opportunity to test the program to ensure that it was performing correctly, but also gave the first independent confirmation of the reported results. During these simulations a number of issues were raised which, after consultation with Nigrin, led to some changes in the network algorithm. Other, more major, modifications are described later.

The question of top-down expectations is not considered in great detail in the description of **SONNET 1**. Since this aspect of the network's performance was of crucial importance to the current application, a means of defining item expectation was developed. Two constraints help to characterize the requirements for a preliminary definition of expectation. Firstly, that the expectation for an item's occurrence should represent the amount of top-down feedback, t_j received by the $\mathcal{F}_1$ cell representing that item; secondly, that items already active at $\mathcal{F}_1$ should not be thought of as being expected. The top-down feedback, t_j, to a given $\mathcal{F}_1$ cell is given by

$$t_j = \sum_{i \in \mathcal{F}_2} d_i z_{ij}^f \tag{6.21}$$

The expectation of a given item, p_j, is thus defined as being equal to zero if the corresponding $\mathcal{F}_1$ cell is active and equal to the top-down feedback signal, t_j, otherwise. The vector formed by the individual item expectations is normalized to allow it to be displayed and interpreted easily. In addition, a number corresponding to the expectation strength (actually the maximum element in the pre-normalization expectation vector) is displayed alongside. It turns out that this simple definition of the top-down feedback, and thus the expectation for a given item, is not sufficiently rigorous to generate the specific, "directed" expectations required by the current application. Steps must be taken to ensure that interference akin to that described in the previous chapter, does not occur. The point is best illustrated with reference to the

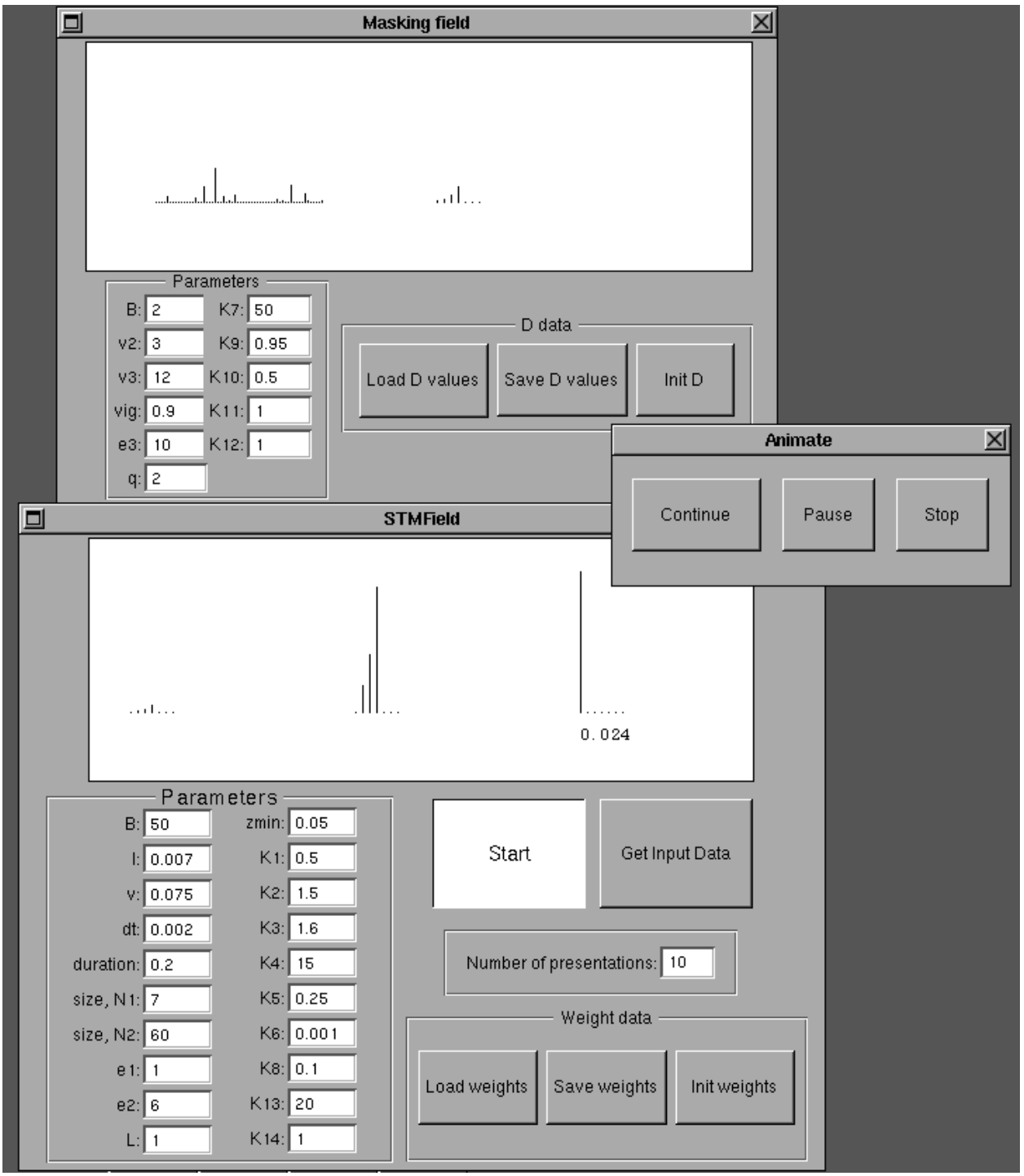


Figure 6.2: A screen dump of the **SONNET 1** simulation

results of simulations, as presented in the next section.

6.5 SONNET 1 simulations

6.5.1 Introduction

The simulations presented in [117] and [118] are mainly concerned with the learning of small numbers of patterns under fast- and slow-learning conditions. They highlight **SONNET 1**'s ability to learn to classify patterns stably under either regime, whilst remaining plastic, i.e. able to learn new patterns as necessary. The effects of variation in certain parameters are illustrated.

The simulations presented here extend those of Nigrin by confronting a **SONNET 1** network with a much larger pattern set consisting of the melodic phrases used in the previous chapter. Since the restrictions on item repetition are not addressed until the next chapter, the "altered" phrases, i.e. those engineered so as to contain no "in phrase" item repetitions, are used here. The phrases themselves are shown again in Table 6.1. The input set was presented to **SONNET 1** in various ways, to establish the capabilities of the network. In all cases, the network is entirely self-organizing and performs unsupervised learning.

The presentation of a given sequence is accomplished in the same manner as that detailed in [117]. Initially the activation of all cells is set to zero. Then for each item in the input sequence in turn, the corresponding external input ($I_i^{(1)}$ in Equation 6.19) is turned on, i.e. set to 1.0, for a given time period. This time period is assumed to be equal to the time period during which changes in activations are enabled at $\mathcal{F}_1$, i.e. the time period for which the attentional signal acts at this level. During this time, changes in all network activations and synaptic weights are similarly enabled. For the case in which a note (event) is prolonged over more than one beat, network changes are enabled for a correspondingly increased time period, with the difference that during this extra period no external input signals are active — this indicates that the network changes are enabled at such times by the attentional signal *in the absence of actual note events.* After the presentation of the last sequence item, network changes are further enabled for a number of "beats", such that the total number of attentional beats is equal to 7 for all sequences presented. For example, for a four-item sequence, the "complete" sequence remains in working memory for a time period equivalent to that required

Phrase	No.of Occurrences	Phrase	No. of Occurrences
1 2 3 7	5	3 5 4 6	1
1 2 5 7	2	4 2 3 1	2
1 3 2 4	4	4 3 2 1	3
1 3 5^2	3	5 3 4 2	2
1 5 3 6	1	5 4 3^2	2
1 5 6 7	3	5 4 3 2	2
1 7 6 5	4	5 4 6 1	2
2 4 5 1	2	5 6 3 1	4
2 5 1^2	1	6 2 3 1	1
2 6 7 5	3	6 5 1^2	3
3 2 1^2	3	6 5 4 3	1
3 4 2 1	4	6 7 1^2	2
3 4 5^2	2		

Table 6.1: The "altered" input set

to present a further three items. This means of presenting "isolated" sequences is consistent with the approach adopted in the next chapter with regard to the "continuous" presentation of items.

6.5.2 The setting of parameters

There are a large number of parameters involved in the definition of the **SONNET 1** network. In many cases consideration of the rôle played by each is sufficient to allow suitable choices to be made—the parameters do not "interact" as much as do those found in the masking field and in **MASKART**, which further facilitates their setting. Nonetheless, preliminary simulations suggested certain settings appropriate to the current application. These are described here.

K_3 and K_4 are involved in the learning of the weights w_{ji}, which are used in the definition of the set T_i. During learning, if the largest value of S_{ji} seen by χ_i is denoted S_{0i}, then values of S_{ji} less than S_{0i}/K_4 are treated as noise. Since we are concerned with sequences, represented by a spatial pattern in working memory for which the shunting parameter, w, is approximately equal to 2, and for which the transient memory span is assumed to be 4, it makes sense to set K_4 in the range $8 < K_4 < 16$. This effectively limits the length of sequences that can be learned at $\mathcal{F}_2$. K_3 sets the criterion for inclusion in the set T_i — it is constrained to be greater than one, and simulations suggested the further constraint that it should not approach 2. In the simulations below, $K_3 = 1.6$ and $K_4 = 12$.

K_7 is important in controlling the speed with which the network can learn patterns similar to previously classified patterns. In the current application we desire the network to learn to differentiate between melodic phrases which are ostensibly very similar, e.g. "anagram" phrases. This requires the setting of K_7 at a value lower than that most often used by Nigrin. In the simulations below, $K_7 = 50$.

The parameter q determines the number of *uncommitted* cells that can respond to a given pattern, and thus *begin* learn to represent that pattern. In order that the size of $\mathcal{F}_2$ (N_2) can be minimized for a given input set (run-time increases approximately proportionally with $(N_2)^2$), q is set small. In the simulations below , $q = 2$.

Nigrin shows that **SONNET 1** is stable for a wide variety of choices for the learning rate, ϵ_2—he uses values in the range $0.1 \leq \epsilon_2 \leq 50$. High learning rates can result in "single-shot" learning, i.e. sequences can be learned in a single presentation; low learning rates require

more presentations for classification, but are more suitable when "embedded" patterns are to be learned. In the simulations below, ϵ_2 was chosen at an intermediate level, $\epsilon_2 = 6$, which, in general, did not allow single-shot learning whilst allowing sequences to be learned in a tractable number of presentations, particularly in "difficult regimes" (see later).

6.5.3 Modifications to the SONNET 1 network

In the course of many simulations, several modifications were made to the **SONNET 1** network. These improved the network's performance in the phrase-learning task, and in some cases were essential to allow "correct" learning in difficult circumstances.

- The first modification involved the working memory, $\mathcal{F}_1$. Initially, activation and weight changes at $\mathcal{F}_1$ and $\mathcal{F}_2$ were enabled, by the attentional signal, over the same time period. This meant that the "full" pattern representing a given sequence, was only present at the very end of the attentional pulse resulting from the onset of its final note. This was undesirable, particularly when the generation of expectations was considered. To address this problem, activation changes at $\mathcal{F}_1$ were enabled for a shorter time than were weight changes and activation changes at $\mathcal{F}_2$. The only assumption is that the attentional signal affects each field differently. The parameters at $\mathcal{F}_1$ were adjusted to ensure that the shunting parameter remained such that $w \approx 2$ for an isochronously presented sequence.

- The second modification involved a change in the way $I^{\times}$ was calculated for committed (or near-committed) cells. Again this resulted from consideration of the network's ability to generate expectations after phrase learning. To illustrate the potential problem, assume that $\chi_{1,2,3,4}$ has learned to represent the sequence $(1,2,3,4)$ and $\chi_{1,2,5,3}$ the sequence $(1,2,5,3)$. On presentation of the sequence $(1,2,3)$, the inputs to each cell are: $I^{+}_{1,2,3,4} = 0.99$, $I^{+}_{1,2,5,3} = 0.97$, $I^{\times}_{1,2,3,4} = I^{\times}_{1,2,5,3} = 4.0$. Thus $b_{1,2,3,4} = 3.96$ and $b_{1,2,5,3} = 3.88$. The small difference in the bottom-up input to each of the cells is at odds with the observation that one of them represents a legitimate superset of the current input, whilst the other does not. In order to remedy this situation, a simple modification was made so that $I^{\times}$ for all cells whose bottom-up weight vector exceeded

a certain length (in the region of 0.7), was calculated thus

$$I_i^{\times} = \max(\prod_{j \in T_i} I_{ji}^{\times}, K_8) \tag{6.22}$$

$$I_{ji}^{\times} = K_1 + K_2 \min(\frac{Z_{ji}}{S_{ji}}, \frac{S_{ji}}{Z_{ji}}) \tag{6.23}$$

Using this formula, the inputs for the case described above would become $I_{1,2,3,4}^{\times} = 4.0$ and $I_{1,2,5,3}^{\times} = 2.5$, thus corresponding more clearly with the "appropriateness" of each cell. Nigrin (personal communication) has also suggested that the formulation for $I^{\times}$ should change as the weights increase at a cell.

- The next modification concerned the inclusion of the learning modulator, r_{ji}, in the differential equation describing the evolution of the "secondary" weights, w_{ji}. During simulations it was found that inclusion of the modulator term could lead to a situation in which the set T_i corresponding to a given $\mathcal{F}_2$ cell (i.e. the set of $\mathcal{F}_1$ cells to which that cell is significantly connected), can shrink in response to shorter sequences (e.g. 3-item sequences), but subsequently is less able to grow again to account for new, longer sequences. The problem is particularly evident when, for instance, a sequence such as $(5, 4, 3)$ has been learned, and the network is presented with the sequence $(5, 4, 3, 2)$. Uncommitted cells, whose set T_i has contracted to {5,4,3} and whose weights are "parallel" to the shorter input pattern, will activate reasonably strongly in response to the longer sequence. The more dominant of these uncommitted cells can not be recruited to learn that sequence if r_{ji} is employed in the w_{ji} learning, since they will have a value of M near unity — this makes learning very slow for connections to $\mathcal{F}_1$ cells not in the set T_i. Note that cells whose $z_i^{tot} < K_5 + K_6$ will not be affected by the inclusion of r_{ji}. To address this problem, the modulator r_{ji} was omitted from the w_{ji} learning equation. This modification proved *essential* in ensuring "correct" performance in the simulations described below.

- A related modification concerned the "shutting off" of uncommitted cells. Nigrin shuts off activity at an uncommitted cell if there are at least q uncommitted cells with a significantly higher e-value and a higher $I^{\times}$. During simulations it was found that the uncommitted cells that remained active in response to a given pattern could have their learning of that pattern slowed by virtue of the fact that their M-values were low. This

was addressed by additionally ensuring that for an uncommitted cell to be shut off, there would have to be at least one other uncommitted cell with a larger e-value and $M > 0.95$. Thus there are always uncommitted cells capable of learning the current input pattern (though obviously this learning can be properly curtailed by lateral inhibition at $\mathcal{F}_2$). This modification also proved essential for correct performance.

In addition a number of smaller modifications were made to the operation of the **SONNET 1** network, such as allowing the D-value of an $\mathcal{F}_2$ cell to continue to grow after that cell has become committed.

6.5.4 Incremental learning of sequences

This series of simulations tested **SONNET 1**'s ability to learn all of the phrases in the input set. A network with 7 $\mathcal{F}_1$ cells and 60 $\mathcal{F}_2$ cells was presented with each of the phrases in turn, in a test of the network's ability to perform *incremental learning*. This regime consisted of the presentation of each phrase as many times as was necessary for an $\mathcal{F}_2$ cell to become committed to classifying that phrase, before proceeding to the presentation of the next phrase. The phrases were presented in four different randomized orders. The variation in the number of presentations (NP) required for the learning of each phrase is illustrated in Table 6.2. This shows how, for each of the random orderings, the network was able to learn incrementally each of the input phrases. Each phrase was learned in a small number of presentations. Moreover, in each case there was no $\mathcal{F}_2$ cell which became "spuriously committed", i.e. committed, for instance, to a subset of one of the input phrases, or to an amalgam of two different phrases. Such learning will be termed *ideal learning*. It was noted that there was a single phrase ($(1,5,3,1)$ in the third ordering of the input set) which required an unusually high number (8) of presentations before being learned. It is difficult to account for this "outlier" — similar behaviour was noted in the simulations presented in the next chapter and a possible reason is suggested there.

The network's performance with this task is encouraging. The input set, whilst only consisting of 25 phrases, contains many phrases which are similar to each other — in particular, there are many "anagram" phrases, such as $(4,3,2,1)$, $(4,2,3,1)$, $(3,4,2,1)$, and $(1,3,2,4)$, which are potentially confusable and which led to problems when presented to **MASKART** (see Section 5.5.1). Another advantage which the **SONNET 1** network enjoys

Order	Phrase	NP	Phrase	NP	Phrase	NP	Phrase	NP
1	345	3	4321	3	5461	4	5432	3
2	2675	3	6543	3	4321	2	1324	2
3	651	2	1536	2	1237	3	543	2
4	1567	2	321	2	251	2	1765	3
5	671	1	345	2	345	2	1257	2
6	6231	2	135	2	2675	3	5631	2
7	3421	2	1257	4	321	3	1536	2
8	1237	3	1237	2	1536	3	4231	3
9	5461	4	5342	2	135	3	2451	3
10	6543	2	6231	4	1765	2	1237	2
11	2451	3	3421	2	5631	8	251	4
12	135	3	251	3	1324	3	135	3
13	4231	3	3546	3	5432	2	321	2
14	4321	4	1567	2	651	3	3421	3
15	1536	2	1324	2	5342	2	6543	3
16	1765	3	5432	4	1257	2	1567	2
17	5342	5	651	1	543	2	4321	5
18	321	4	671	4	1567	2	671	4
19	1257	2	675	3	6231	3	345	4
20	543	2	5631	3	671	2	5461	2
21	251	2	5461	2	4231	4	2675	4
22	1324	3	1765	3	3421	2	651	2
23	5432	3	2451	1	3546	3	3546	4
24	5631	2	4231	3	6543	3	5342	5
25	3546	2	543	3	2451	2	6231	2

Table 6.2: Number of presentations (NP) needed to learn each phrase for four different random orderings (incremental learning)

over **MASKART**, is the comparatively small number of $\mathcal{F}_2$ cells required to permit learning of the input set. For **MASKART** we used 336 $\mathcal{F}_2$ cells in an equivalent learning task, and even this number was kept lower than would have been necessary to learn an "arbitrary" input-set. By contrast, the **SONNET 1** network used above had only 60 $\mathcal{F}_2$ cell assemblies. The fact that **SONNET 1**'s inhomogeneity emerges in response to a given input set, rather than being "hard-wired", permits the "compression ratio", defined as the number of $\mathcal{F}_2$ cells used in achieving ideal learning divided by the number of distinct input patterns, to remain small.

6.5.5 Learning with "cyclic" presentation

The simulations described above showed how the modified **SONNET 1** network performed in an incremental-learning regime. In such a regime the network is only being expected to learn a single phrase at any given time— this perhaps renders the task unrealistically easy. Accordingly, a much more challenging regime was employed, under which the network was "cyclically" exposed to the entire input set (again arranged in a random ordering). In these simulations, therefore, consecutive presentations of any given phrase are separated by the presentation of each of the other phrases. The simulations employing cyclic presentation presented in [117]and [118], use either a small number (2) of overlapping sequences (e.g. AB and BA) or a slightly larger set (8) of non-overlapping patterns. The simulations described here used a much larger set (25) of "highly-overlapping" patterns, and thus constitute a stern test of the network's performance. It was during these simulations that many of the modifications to the network were made. As noted above, without these modifications ideal learning could not be achieved — the most common error was the learning of subphrases, e.g. the learning of the subphrase $(1, 5, 3)$ in response to the phrase $(1, 5, 3, 6)$. Such errors did not manifest themselves in the incremental-learning regime, even for the unmodified network.

The performance of the modified **SONNET 1** network, with cyclic presentation of the input-set and the same learning-rate as was used for the incremental learning regime, is illustrated in Figure 6.3. The first phrase to be learned is learned on the very first presentation; 47 presentations of the complete input-set are necessary before all phrases are learned. Nonetheless, even in this most difficult of learning regimes, ideal learning is achieved. This means of presentation also illustrates the stability of the network's classifications, in particular its

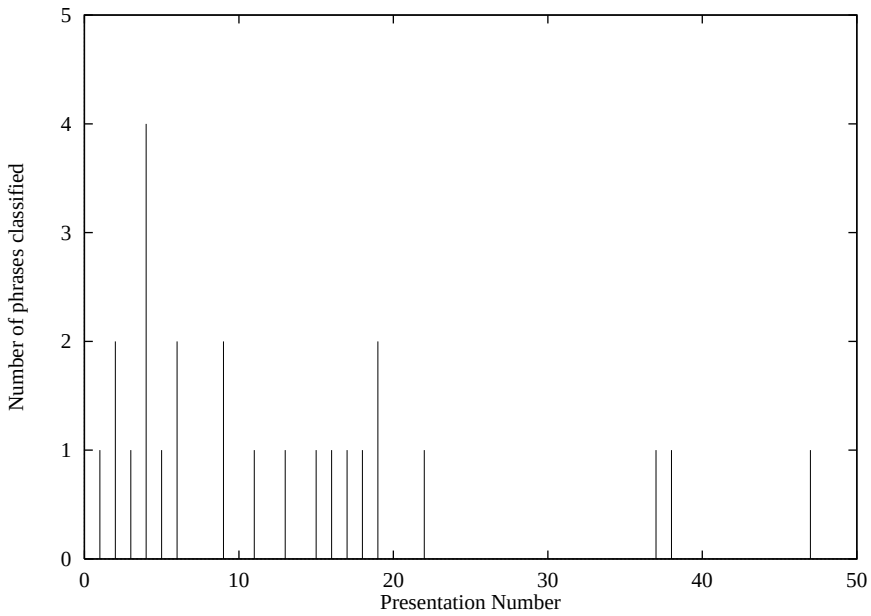


Figure 6.3: Result for cyclic presentation of the input set

ability to prevent multiple cells from classifying the same pattern. Table 6.3 shows the number of presentations required to learn each specific phrase. It is notable that the learning of the shorter (three-note) phrases tend to require more presentations than that of the four-note phrases: the mean number of presentations required to learn a three-note phrase was 27(s.d. 14); the mean number of presentations required to learn a four-note phrase was 8 (s.d. 6.3). This disparity is attributable to the modification made with regard to the influence of the learning modulator, r_{ji}, on the evolution of the secondary weights, w_{ji}. This modification was made expressly to prevent an $\mathcal{F}_2$ cell's set T_i from shrinking too fast, a situation which led, in earlier simulations, to the learning of sub-phrases. The effect of this modification on the learning of short phrases in a cyclic learning-regime, is the price paid to ensure ideal learning.

It should be noted that cyclic learning was only employed for a single random ordering—the main constraint was the long run-time required to perform such simulations, including those simulations which proved unsuccessful and led to network modifications. Nonetheless, the success of the modified **SONNET 1** network in this difficult learning-regime was encouraging. The next section describes how "plausible" expectations can be elicited from a network that has learned, under whatever regime, the complete input-set.

Order	Phrase	NP	Order	Phrase	NP
1	345	47	14	4321	9
2	2675	17	15	1536	18
3	651	13	16	1765	4
4	1567	15	17	5342	4
5	671	22	18	321	37
6	6231	5	19	1257	19
7	3421	3	20	543	11
8	1237	2	21	251	38
9	5461	6	22	1324	1
10	6543	19	23	5432	2
11	2451	4	24	5631	9
12	135	18	25	3546	6
13	4231	4			

Table 6.3: Number of presentations (NP) needed to learn each phrase with cyclic learning

6.6 Modifying the conditions for top-down feedback

Once a **SONNET 1** network has learned a set of input patterns, in this case corresponding to a set of short melodic phrases, we can "probe" it with sub-phrases to investigate the expectations for phrase-completion that are elicited. In the previous chapter, it was shown that the **MASKART** network, even after "ideal" learning of the input set, was not able to generate expectations that constituted plausible continuations of sub-phrases. This failure was attributed to "interference" from the broad set of cells that typically activated in response to a probe sequence—the expectations were not sufficiently well "directed". It was further shown that it was possible to obtain an acceptable level of performance if the set of cells able to generate top-down priming signals was restricted to those cells that had learned to represent entire phrases, rather than just phrase parts. Unfortunately, it was only possible to define this set of cells "manually", i.e. by interference in **MASKART**'s behaviour. This section will show that **SONNET 1** also suffers from interference effects in the generation of priming signals, but that these can be eliminated using only information internal to the network. This makes the **SONNET 1** network more suitable for application as a sequential memory. All the expectations listed here were generated by probing a single **SONNET 1** network that had been exposed to, and learned, the set of phrases shown in Table 6.1.

Tables 6.4-6.6 show the response of the network to the set of one-, two- and three-note probes derived from the input set. As in the previous chapter, the expectations for each of the seven possible input items are listed and the "validity" of the network's expectations is summarized with a code as before. The "correct" expectation is gauged by looking at the learned phrase-set, without taking into account the relative frequency of various continuations, e.g. suppose the network has learned three four-note phrases beginning $(1,2)$ and two four-note-phrases beginning $(1,3)$, the response deemed correct for a probe of (1) would be one which indicated that either a "2" or a "3" might come next, without significantly weighting the expectation towards a "2". This means of assessing *possible* continuations, rather than just the most common continuation (defined here as the number of learned phrases that contain that continuation) is more appropriate to the use of **SONNET 1** as a veridical sequential memory — this point will be discussed later with reference to the calculation of the top-down feedback signals.

As can be seen, interference once again disrupts the generation of plausible expectations.

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1	0.000	0.025	0.064	0.008	0.044	0.019	0.012	$*$
2	0.033	0.000	0.023	0.013	0.030	0.009	0.004	$*$
3	0.070	0.036	0.000	0.044	0.040	0.006	0.003	$\circ$
4	0.015	0.020	0.041	0.000	0.029	0.002	0.000	$*$
5	0.035	0.025	0.042	0.044	0.000	0.029	0.002	$*$
6	0.029	0.010	0.006	0.004	0.035	0.000	0.025	$*$

Table 6.4:

For instance, after presentation of the phrase $(5, 4, 3, 2)$, there is a strong expectation for the item "6", in spite of the fact that there is never a "6" found in such a position in the input-set. The expectation for the "6" constitutes a "back-expectation", which results from the activation of the cell $\chi_{6,5,4,3}$. Eliminating such interference requires restricting the set of $\mathcal{F}_2$ cells that can generate top-down priming signals, so as to ensure that only genuine continuations, consistent with past experience, are reflected in the pattern of expectations.

The restrictions placed on the set of expectation-generating $\mathcal{F}_2$ cells can be described in several stages:

- Firstly, we can stipulate that only $\mathcal{F}_2$ cells that have become committed to an input pattern can generate top-down priming signals. This seems a reasonable heuristic — a given continuation will only be expected if it forms part of a phrase that has occurred frequently enough to become represented at $\mathcal{F}_2$. This measure also takes into account the fact that there will *always* be a small number of uncommitted cells active at $\mathcal{F}_2$ (by design) and that these might well otherwise generate spurious expectations.

- Secondly, we must allow some differentiation between cells which are "on-target" and those which are not: a cell which is on-target is defined as one whose learned pattern is entirely consistent with all, or part, of the current input ($\mathcal{F}_1$) pattern. Thus, if the input pattern represents the sequence $(1, 2, 3)$, then the $\mathcal{F}_2$ cell $\chi_{1,2,3,4}$ is on-target, whilst the cell $\chi_{1,3,2,4}$ is not. Likewise, for the input pattern representing the sequence $(5, 4, 3)$, the cells $\chi_{5,4,3}$ and $\chi_{5,4,3,2}$ are on-target, whereas the cell $\chi_{6,5,4,3}$ is not. In order to

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1,2	0.000	0.000	0.192	0.031	0.114	0.045	0.034	$*$
1,3	0.000	0.147	0.000	0.054	0.165	0.039	0.027	$*$
1,5	0.000	0.104	0.181	0.046	0.000	0.084	0.035	$*$
1,7	0.000	0.069	0.115	0.008	0.108	0.080	0.000	$\times$
2,4	0.100	0.000	0.093	0.000	0.092	0.019	0.007	$\times$
2,5	0.130	0.000	0.069	0.069	0.000	0.051	0.011	$*$
2,6	0.086	0.000	0.047	0.028	0.095	0.000	0.040	$\times$
3,2	0.217	0.000	0.000	0.116	0.107	0.025	0.009	$*$
3,4	0.138	0.084	0.000	0.000	0.169	0.017	0.006	$*$
3,5	0.170	0.086	0.000	0.153	0.000	0.043	0.006	$\circ$
4,2	0.090	0.000	0.141	0.000	0.087	0.014	0.004	$*$
4,3	0.064	0.081	0.000	0.000	0.118	0.009	0.001	$\circ$
5,3	0.146	0.093	0.000	0.166	0.000	0.072	0.006	$*$
5,4	0.061	0.074	0.173	0.000	0.000	0.069	0.002	$*$
5,6	0.105	0.060	0.093	0.094	0.000	0.000	0.030	$\times$
6,2	0.083	0.000	0.042	0.026	0.095	0.000	0.049	$\times$
6,5	0.122	0.056	0.054	0.056	0.000	0.000	0.048	$*$
6,7	0.116	0.029	0.014	0.007	0.067	0.000	0.000	$*$

Table 6.5:

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1,2,3	0.000	0.000	0.000	0.106	0.219	0.086	0.066	×
1,2,5	0.000	0.000	0.335	0.076	0.000	0.091	0.079	×
1,3,2	0.000	0.000	0.000	0.130	0.234	0.061	0.044	∘
1,5,3	0.000	0.194	0.000	0.142	0.000	0.152	0.052	∘
1,5,6	0.000	0.162	0.299	0.086	0.000	0.000	0.102	×
1,7,6	0.000	0.100	0.141	0.015	0.193	0.000	0.000	∗
2,4,5	0.201	0.000	0.191	0.000	0.000	0.068	0.016	∗
2,6,7	0.186	0.000	0.068	0.040	0.159	0.000	0.000	∘
3,4,2	0.239	0.000	0.000	0.000	0.309	0.030	0.008	∘
3,5,4	0.232	0.136	0.000	0.000	0.000	0.106	0.007	×
4,2,3	0.189	0.000	0.000	0.000	0.202	0.032	0.008	∘
4,3,2	0.208	0.000	0.000	0.000	0.240	0.027	0.006	∘
5,3,4	0.202	0.171	0.000	0.000	0.000	0.122	0.006	∘
5,4,3	0.107	0.147	0.000	0.000	0.000	0.085	0.003	∗
5,4,6	0.147	0.128	0.304	0.000	0.000	0.000	0.029	∘
5,6,3	0.224	0.130	0.000	0.236	0.000	0.000	0.051	∘
6,2,3	0.195	0.000	0.000	0.115	0.194	0.000	0.067	∗
6,5,4	0.165	0.081	0.191	0.000	0.000	0.000	0.056	∗

Table 6.6:

distinguish between cells in this way, we take advantage of the fact, previously noted and regarded as a problem, that the value of I^+ is insensitive to the presence/absence of items towards the end of its list. Cells that are on-target tend to have a larger value of I^+ than those which are not. Using the examples given above and assuming perfect learning of patterns generated by a working memory with shunting parameter $w = 2$: on presentation of the sequence $(1, 2, 3)$, $I^+_{1,2,3,4} = 0.994$ and $I^+_{1,3,2,4} = 0.947$; on presentation of the sequence $(5, 4, 3)$, $I^+_{5,4,3} = 1.0$, $I^+_{5,4,3,2} = 0.994$ whereas $I^+_{6,5,4,3} = 0.497$. Thus we can restrict the set of cells that are able to generate top-down priming signals to those $\mathcal{F}_2$ cells for which $I^+ > K_{15} I^+_{max}$, where I^+_{max} is the maximum value of I^+ currently found in $\mathcal{F}_2$. The setting of the constant K_{15} is critical: the limiting case, one which is found when considering the current input set, is when the sequence $(1, 5, 6)$ is presented to a network that has learned both the sequences $(1, 5, 6, 7)$ and $(1, 5, 3, 6)$, at which point $I^+_{1,5,6,7} = 0.994$ and $I^+_{1,5,3,6} = 0.9705$. In order, therefore, that the on-target distinction is correctly made, K_{15} must be greater than 0.9705. If, as above, we want both $\chi_{5,4,3}$ and $\chi_{5,4,3,2}$ to be deemed to be on-target on presentation of the item "5" or the sequences $(5, 4)$ and $(5, 4, 3)$, then we must set K_{15} to be less than 0.994. A value of $K_{15} \approx 0.98$ is suitable.

- The final restriction on the set of cells able to generate top-down signals can once again be illustrated with an example drawn from the current input set: on presentation of the sequence $(5, 3, 4, 2)$, $I^+_{5,3,4,2} = 1.0$ and $I^+_{3,4,2,1} = 0.994$, implying that both cells will be considered on-target and able to generate expectations. This will lead to the expectation for a "1", even though the phrase is both familiar and complete. A similar problem occurs for the phrases $(3, 5, 4, 6)$ and $(5, 4, 6, 1)$ etc.. Two possible approaches were considered to avoid this problem. Firstly, it was postulated that if any cell reaches its classification threshold, K_{11}, then that cell alone will be able to generate top-down signals (but not *expectations* since its full pattern will be present at $\mathcal{F}_1$). This measure provided a partial solution (as well as fitting well with the "hierarchical priming" described in the next chapter), but was not guaranteed to suppress unwarranted expectations since the activation of the cell occasionally falls just short of K_{11} by the time the next "beat" occurs (by which time expectations should be resolved). The second solution, which can be employed in addition to the first rather than in place of it, involved placing a

restriction on $I^{\times}$ similar to that imposed on I^{+}: only cells for which $I^{\times} > K_{16} I^{\times}_{max}$ can generate top-down signals, where $I^{\times}_{max}$ is the maximum value of $I^{\times}$ currently found in $\mathcal{F}_2$. Considerations similar to those above suggest the setting $K_{16} \approx 0.45$. It is worth noting that the stipulations put on the values of both I^{+} and $I^{\times}$ cannot be collapsed into a single stipulation on the value of b (where $b = I^{+}I^{\times}$), since simulations have shown that problems can result when a relatively high value of $I^{\times}$ compensates for a relatively low value of I^{+}.

There is one more modification that must be made to the means by which the network generates top-down priming signals. As noted earlier, we wish the network to provide expectations of *possible* continuations without regard to the number of different learned phrases with which those expectations are consistent, since there is no reason to believe that this information necessarily reflects the *probability* of a given transition. The top-down feedback signal, t_j, received by the j^{th} $\mathcal{F}_1$ cell, was previously calculated by

$$t_j = \sum_{i \in \mathcal{F}_2} d_i z^f_{ij} \tag{6.24}$$

We now calculate the top-down priming with

$$t_j = \max_{i \in P}(d_i z^f_{ij}) \tag{6.25}$$

where P is the set of $\mathcal{F}_2$ cells permitted to generate top-down signals. Thus instead of summing the priming signals from individual $\mathcal{F}_2$ cells, we now take the *maximum* top-down feedback signal experienced at a given $\mathcal{F}_1$ cell. The values for the expectation, p_j for a given item are calculated from the values t_j as before.

Having restricted the set of $\mathcal{F}_2$ cells that are able to generate top-down priming signals, and having modified the calculation of t_j, the same network as was used above, was once again probed with the one-, two- and three-note probe sequences. The expectations elicited are shown in Tables 6.7-6.9 and are assessed as before. The improvement in performance is dramatic: for each of the probe sequences, the network's expectations correspond exactly with those suggested by inspection of the complete input pattern set. In addition, full presentation of a learned phrase resulted in zero expectation in all cases, apart from presentation of the phrase $(5, 4, 3)$ which elicited an expectation for the item "2". A larger context is necessary in order to determine the circumstances under which this expectation is "correct" — this

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1	0.000	0.006	0.023	0.001	0.011	0.003	0.006	*
2	0.011	0.000	0.000	0.006	0.024	0.006	0.003	*
3	0.010	0.022	0.000	0.023	0.011	0.000	0.000	*
4	0.001	0.006	0.006	0.000	0.000	0.000	0.000	*
5	0.001	0.000	0.011	0.023	0.000	0.005	0.000	*
6	0.011	0.000	0.000	0.000	0.023	0.000	0.023	*

Table 6.7:

issue is dealt with in the next chapter. It should be noted that in the situation in which two cells, one representing a three-note phrase the other a four-note phrase, are both on-target in response to a one- or two-note probe, then the expectations generated by the cell representing the three-note phrase will dominate those generated by the cell representing the four-note phrase — this is consistent with the fact that a greater proportion of the three-note phrase is present at the $\mathcal{F}_1$ layer. In the next chapter, it will be shown how this effect can be overcome (as it must be in certain circumstances) by the introduction of hierarchical expectations.

There are a number of drawbacks associated with this means of defining the set P so as to control top-down priming. The most important concerns the setting of the parameter K_{15}. The range of values for which K_{15} fulfills its desired function is narrow ($0.977 \leq K_{15} < 0.994$ assuming "perfect learning") and its operation might therefore be adversely affected by any "noise" in the system. Secondly, the expectations from $\mathcal{F}_2$ cells that represent short sequences, e.g. two-note sequences, will exclude those from cells representing larger, superset sequences — this problem did not arise here since the input set does not include any two-note phrases. The final drawback involves the additional network structure required to "broadcast" the value of I^+_{max} to all cells in $\mathcal{F}_2$ — though this is easily supplied it does add complexity to an already complex network. The next section describes a more fundamental modification to the network, which addresses these drawbacks.

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1,2	0.000	0.000	0.015	0.000	0.014	0.000	0.007	*
1,3	0.000	0.014	0.000	0.006	0.054	0.000	0.000	*
1,5	0.000	0.000	0.015	0.000	0.000	0.014	0.007	*
1,7	0.000	0.000	0.000	0.000	0.007	0.015	0.000	*
2,4	0.007	0.000	0.000	0.000	0.014	0.000	0.000	*
2,5	0.057	0.000	0.000	0.000	0.000	0.000	0.000	*
2,6	0.000	0.000	0.000	0.000	0.007	0.000	0.014	*
3,2	0.053	0.000	0.000	0.000	0.000	0.000	0.000	*
3,4	0.007	0.016	0.000	0.000	0.057	0.000	0.000	*
3,5	0.000	0.000	0.000	0.014	0.000	0.007	0.000	*
4,2	0.007	0.000	0.014	0.000	0.000	0.000	0.000	*
4,3	0.007	0.015	0.000	0.000	0.000	0.000	0.000	*
5,3	0.000	0.007	0.000	0.015	0.000	0.000	0.000	*
5,4	0.007	0.007	0.055	0.000	0.000	0.015	0.000	*
5,6	0.006	0.000	0.014	0.000	0.000	0.000	0.000	*
6,2	0.007	0.000	0.014	0.000	0.000	0.000	0.000	*
6,5	0.055	0.000	0.007	0.015	0.000	0.000	0.000	*
6,7	0.055	0.000	0.000	0.000	0.000	0.000	0.000	*

Table 6.8:

Probe	Expectations							Code
	1	2	3	4	5	6	7	
1,2,3	0.000	0.000	0.000	0.000	0.000	0.000	0.027	*
1,2,5	0.000	0.000	0.000	0.000	0.000	0.000	0.027	*
1,3,2	0.000	0.000	0.000	0.024	0.000	0.000	0.000	*
1,5,3	0.000	0.000	0.000	0.000	0.000	0.026	0.000	*
1,5,6	0.000	0.000	0.000	0.000	0.000	0.000	0.026	*
1,7,6	0.000	0.000	0.000	0.000	0.027	0.000	0.000	*
2,4,5	0.027	0.000	0.000	0.000	0.000	0.000	0.000	*
2,6,7	0.000	0.000	0.000	0.000	0.026	0.000	0.000	*
3,4,2	0.028	0.000	0.000	0.000	0.000	0.000	0.000	*
3,5,4	0.000	0.000	0.000	0.000	0.000	0.025	0.000	*
4,2,3	0.026	0.000	0.000	0.000	0.000	0.000	0.000	*
4,3,2	0.027	0.000	0.000	0.000	0.000	0.000	0.000	*
5,3,4	0.000	0.026	0.000	0.000	0.000	0.000	0.000	*
5,4,3	0.000	0.019	0.000	0.000	0.000	0.000	0.000	*
5,4,6	0.026	0.000	0.000	0.000	0.000	0.000	0.000	*
5,6,3	0.025	0.000	0.000	0.000	0.000	0.000	0.000	*
6,2,3	0.027	0.000	0.000	0.000	0.000	0.000	0.000	*
6,5,4	0.000	0.000	0.027	0.000	0.000	0.000	0.000	*

Table 6.9:

6.6.1 The calculation of W^+ and $W^\times$

The measures for restricting the set of $\mathcal{F}_2$ cells, P, capable of generating top-down priming signals, described in the previous section, were those adopted with the aim of using only the network information currently available, i.e. $I^+, I^\times$ etc.. This restriction was dropped in the light of the drawbacks discussed in the previous paragraph. I propose the definition of two additional variables, W^+ and $W^\times$, whose calculation is analogous to that of I^+ and $I^\times$, but which employs the "secondary" weights, w_{ji}, rather than the weights z_{ji}. The new variables are defined thus

$$W_i^+ = \sum_{j \in T_i} W_{ji}^+ \tag{6.26}$$

$$W_{ji}^+ = \begin{cases} 1 & \text{if } K_4 S_{ji} > S_{0i} \text{ and } S_{0i} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6.27}$$

$$W_i^\times = \max(\prod_{j \in T_i} W_{ji}^\times, K_8) \tag{6.28}$$

$$W_{ji}^\times = \begin{cases} K_1 + K_2 & \text{if } K_4 S_{ji} > S_{0i} \text{ and } S_{0i} > 0 \\ K_1 & \text{otherwise} \end{cases} \tag{6.29}$$

where S_{0i} is defined as before. As is noted above, the equations for W^+ and $W^\times$ are calculated in a way that is entirely analogous to the calculation of I^+ and $I^\times$. Thus, even though new *variables* are being introduced, no novel mechanism is introduced to account for their calculation. In simple terms, for a given $\mathcal{F}_2$ cell, the value of W^+ represents the number of $\mathcal{F}_1$ cells that are in the set T_i and that have significant activation (defined as before); the value of $W^\times$ represents the highest value of $I^\times$ that a cell can be expected to have as a result of the currently active $\mathcal{F}_1$ pattern.

The newly-defined variables can now be used in the definition of the set, P, of $\mathcal{F}_2$ cells which can generate top-down priming signals. We can ascertain those cells which are on-target by comparing the values of $I^\times$ and $W^\times$ at a cell. Since $W^\times$ represents the highest value of $I^\times$ that can be expected at a given $\mathcal{F}_2$ cell *given the current input pattern*, a cell can be deemed to be on-target if the actual value of $I^\times$ is not significantly lower than this. Appropriating the constant K_{15}, whose former application now becomes irrelevant, we can say that an $\mathcal{F}_2$

cell is on-target if

$$I^{\times} > K_{15} W^{\times} \tag{6.30}$$

where considerable latitude is available in the setting of K_{15}: it can take any value such that $0.7 \leq K_{15} \leq 0.9$ without affecting its function. Thus by a purely local comparison, we can ascertain which cells are on-target, with the additional advantage that *all* cells that are on-target can be discerned in this way, regardless of their size (remember that this was not possible before if short sequences were represented at $\mathcal{F}_2$).

There remains the problem of unwarranted expectations that can occur, for instance, in response to the sequence $(1, 2, 3, 4)$, if there exist committed $\mathcal{F}_2$ cells $\chi_{1,2,3,4}$ and $\chi_{3,4,5}$. Using the definition above, both cells will be considered on-target, leading to the expectation of a "5" in spite of the apparent completion of a phrase. We can overcome this problem by comparing the values of W^+ for the two cells. In the circumstances described, $W^+_{1,2,3,4} = 4$ whilst $W^+_{3,4,5} = 2$. Thus, in order to define properly the set P, we conclude that a cell not only has to be committed and on-target, but that it also must satisfy

$$W^+ > K_{16} W^+_{max} \tag{6.31}$$

where W^+_{max} is the maximum value of W^+ found at an on-target cell. This comparison of W^+ requires the "distributing" of this information across $\mathcal{F}_2$, and thus some additional network structure; the comparison is not susceptible to noise, however, since W^+ is "quantized", i.e. it is only able to take integer values. This allows latitude in the setting of K_{16}, which can be set at any value such that $0.8 \leq K_{16} < 1.0$.

This method of defining the set P doesn't suffer any of the major drawbacks associated with the previous method and is therefore preferable, in spite of the additional calculation required. It should be noted that if there is no activity at $\mathcal{F}_1$, then all committed $\mathcal{F}_2$ cells will be in the set P. This is crucial for allowing the "hierarchical expectations", described in the next chapter, to make themselves properly evident.

The value of W^+_i can also be used in other ways: for instance, in section 6.2.6, the condition $I^{\times}_j \approx K_1^{|T_j|}$ was used to indicate that none of χ_j's pattern is present at $\mathcal{F}_1$; this condition can now be replaced with $W^+_j = 0$.

6.6.2 "Chunking out"

There is a further subtlety concerning the way in which patterns are chunked out of working memory. Nigrin postulates that when an $\mathcal{F}_2$ cell has been highly active ($c_i > K_{11}$) for a time K_{12}, then its activity is reset (and is "forwarded" to a higher-level), causing an abrupt drop in the top-down feedback to the $\mathcal{F}_1$ cells in its receptive field. $\mathcal{F}_1$ cells that experience such an abrupt drop in top-down feedback are themselves reset. This mechanism can, however, lead to problems when the top-down priming is restricted in the way described above: the feedback from an $\mathcal{F}_2$ cell can now be abruptly halted when that cell is removed from the set, P, of cells permitted to generate top-down signals, even though the cell itself remains active. It is therefore necessary to postulate that $\mathcal{F}_1$ cells are reset directly, i.e. by the sending of a specific reset signal to all cells in the receptive field of a given $\mathcal{F}_2$ cell, when that cell reaches the "classification threshold". This direct resetting of $\mathcal{F}_1$ cells also brings advantages when we consider the situation in which multiple cells can respond to the same input item or pattern — this is dealt with in the next chapter.

6.7 Summary

This chapter has described Nigrin's **SONNET 1** network. The first independent simulations of that network have been discussed, together with the modifications that these suggested. The modified network proved able to learn "ideally" a set of patterns representing short melodic phrases, under both incremental and cyclic presentation regimes. A robust means of eliciting plausible expectations from the resulting network has been introduced and verified by simulation.

Chapter 7

Extensions to SONNET 1

7.1 Introduction

In the previous chapter I introduced Nigrin's **SONNET 1** network and showed how, after certain modifications had been made, it could be applied to the unsupervised learning of a set of short melodic phrases. Having learned these phrases, it was shown that, by placing constraints on the set of $\mathcal{F}_2$ cells that were able to generate top-down feedback signals, the network was able to generate plausible expectations when probed with incomplete phrases. Two major problems with the formulation of **SONNET 1** were highlighted: firstly, its inability to handle phrases containing repeated items, a deficiency which necessitated the presentation of isolated phrases engineered so as not to contain such repetitions; and secondly, the lack of any hierarchical structure to enable the unambiguous generation of specific, veridical expectations given sufficient context. Both of these problems are noted in [117] and solutions are proposed: the general ideas underlying the building of a hierarchy are discussed, but no simulation results accompany the description; likewise, a potential solution to the problem of repeated items is discussed but was not implemented (personal communication). This chapter describes in detail the implementation of both proposed solutions and the application of the resulting network to melody learning. Since the problem of repeated items so fundamentally restricts the operation of the network, this will be dealt with first.

7.2 Dealing with repeated items

7.2.1 Proposed solution

Before discussing implementation details, I shall briefly review the considerations which led to Nigrin's proposed solution—the full discussion of these, and other related issues, can be found in Chapter 8 of [117]. For compatibility with this full description, I shall refer to the processing of letter sequences, e.g. ABCD, rather than note sequences.

Nigrin's first conclusion is that one cannot, in the current framework, represent multiple occurrences of an item by the activation of a single cell, since item-order information could not thus be conveyed. There must, therefore, be "multiple locations where the item can be stored". He further concludes that it is not practical to tag each location with an index indicating the "occurrence-number" of the relevant item: such a scheme would store the sequence BABC by activating, in order, the distinct $\mathcal{F}_1$ cells B_1, A_1, B_2 and C_1, where the subscript indicates the occurrence-number of the item. The problem lies in the processing of continuous streams of items: when presented with the sequence ABCADE, should the network treat the second occurrence of item A as the second A in the list ABCA, in which case it should activate cell A_2, or as the first A in the list ADE, in which case it should activate the cell A_1 (at least once the preceding list ABC, if familiar, has been chunked out)? Clearly, in the latter case, when the list ABC is chunked out, we would want the activation of cell A_2 to "transfer" to cell A_1 — this is impractical. Thus each location at which item A can be represented must be able to represent A "without regard to its position in the list".

Nigrin points out that the division of an $\mathcal{F}_2$-type field into a number of "competitive clusters", such that cells in a given cluster only compete with (i.e. inhibit) cells in the same cluster and not cells in other clusters, will give rise to a situation in which multiple cells (in fact as many cells as there are clusters) will come to represent any given, frequently-presented pattern. He suggests that when a pattern is classified, i.e. when the activity, c, of the relevant $\mathcal{F}_2$ cell exceeds K_{11} for a time K_{12}, a corresponding "latch" cell, located in a field which is intermediate between $\mathcal{F}_2$ and the $\mathcal{F}_1$ field of a "higher-order" module, is activated. Whereas multiple cells in the $\mathcal{F}_2$ field can activate in response to the pattern (which may be a low-level, distributed representation of, say, item A), only one cell in the latch field is permitted to be active at any given time, i.e. the latch field is not divided into competitive clusters but is

fully inhibitory. The active latch cell stimulates a corresponding (one-to-one) s-cell in the $\mathcal{F}_1$ field above, in turn implying that there exist multiple s-cells in that field that respond to the same item. Finally, s-cells in a given $\mathcal{F}_1$ field inhibit the activation of the corresponding c-cells in the $\mathcal{F}_2$ layer below. This prevents latch cells that correspond to active s-cells from firing, so that the repetition of a recently presented item will automatically activate one of the alternative s-cells that can represent that item. The arrangement is shown in Figure 7.1. Suppose that we allow four competitive clusters to exist at each $\mathcal{F}_2$ layer — this implies that, at the $\mathcal{F}_1$ layer above, there will be four s-cells which can activate in response to a given item, say item A, only one of which can be receiving input from its latch cell at a given time. These four cells can be labelled A_1, A_2, A_3 and A_4, where the subscripts do not now serve to distinguish between occurrence-number or list-position, but simply allow us to distinguish between the four cells. Now suppose that the network has been exposed to, and has learned, the sequence AAAA. It is clear that this sequence must be recognized whatever the order in which it activates the corresponding $\mathcal{F}_1$ cells, i.e. regardless of whether the sequence activates the cells in the order A_1 A_2 A_3 A_4, or in the order A_2 A_1 A_3 A_4 etc.. To avoid the combinatorial explosion that would result if we insisted that a new $\mathcal{F}_2$ cell were necessary to learn one of each of the 24 different activation-orderings, it is postulated that a single $\mathcal{F}_2$ cell, labelled χ_{AAAA}, is able to learn to respond to the sequence whatever the activation-ordering. This forces a major modification in the network connectivity — instead of there being a single connection from each of the A_n to χ_{AAAA}, there must now be multiple (in this case four) connections. The weighting of these connections must be such that between any given cell A_n and the $\mathcal{F}_2$ cell χ_{AAAA} there will be four connections of different weights, each weight corresponding to one of the four different list positions in which the item can appear.

Nigrin suggests a further major modification: in addition to cells being able to inhibit other cells, links (connections) between cells are also able to inhibit other links. In fact, if we postulate that links are made via "relay" cells, then the distinction between cells and links becomes unimportant. This modification allows us to specify two conditions on the inhibition required between links, to allow the network to function as desired:

- All links that share the same terminal cells (one in $\mathcal{F}_1$, the other in $\mathcal{F}_2$) should be mutually inhibitory, so that *only one* of the multiple links between those two cells is "active" at any given time.

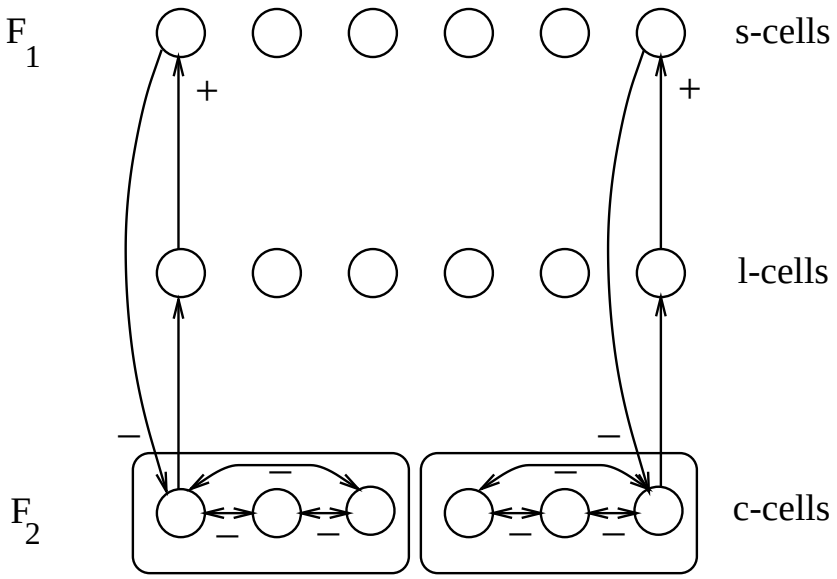


Figure 7.1: The arrangement of c-, l- and s-cells.

- The link from a given $\mathcal{F}_1$ cell, say A_i, to the $\mathcal{F}_2$ cell χ_j should inhibit all *corresponding* links from other cells A_k $(k \neq i)$ to χ_j, i.e. all links that "code" the item in the same list position. Moreover, the weights on *corresponding* links should be kept the same, i.e. learning that is carried out at any one link should be identically carried out at all corresponding links.

Figure 7.2 illustrates these conditions, showing the connectivity required between the two $\mathcal{F}_1$ cells A_1 and A_2 and the $\mathcal{F}_2$ cell χ_{AA}. When one of the $\mathcal{F}_1$ cells, say A_1, activates, its stronger link to χ_{AA} will suppress the weaker link, in addition to suppressing the stronger link emanating from A_2. When A_2 activates, therefore, its weaker link will be functional. By contrast, if A_2 activates first, its stronger link will be enabled, followed by A_1's weaker link. Nigrin shows how this pattern of connectivity can self-organize, though his explanation relies on the assumption that an $\mathcal{F}_2$ cell is able discern that two of its incoming links activate in response to the same item. Self organization of the inhibitory connections is not simulated by Nigrin, and the desired connectivity pattern is, likewise, assumed here.

7.2.2 Implementation

The design of a computer program to simulate the behaviour of a network configured as described above entails certain compromises in the name of computational efficiency. The most obvious of these regards the division of $\mathcal{F}_2$ into competitive clusters: this division is

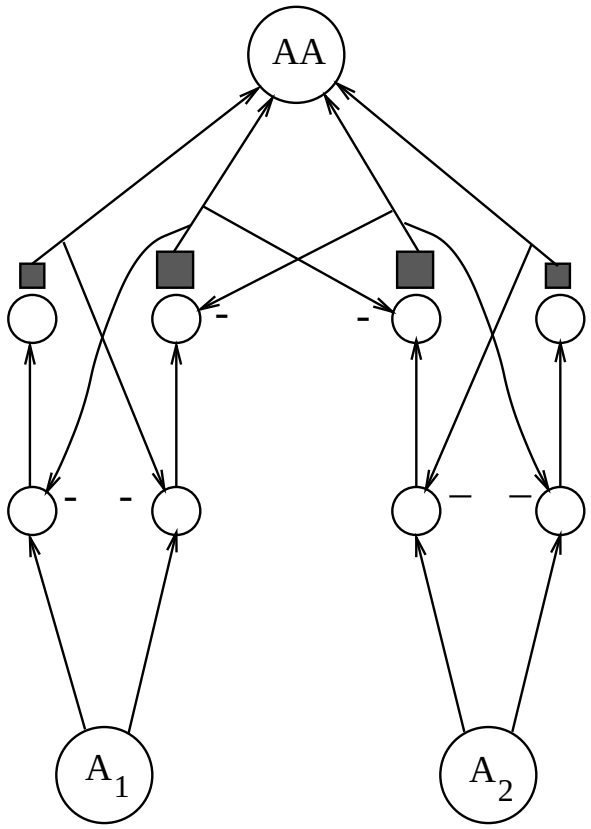


Figure 7.2: The pattern of inter-link inhibition.

required to enable multiple cells to respond identically to a given familiar $\mathcal{F}_1$ pattern. The fact that they respond identically, however, negates the need to simulate separately the changes in activity within each of the clusters — we can simply simulate the behaviour of a single cluster and assume that the other clusters behave identically. Although this assumption is not strictly justified, since, for example, the random initialization of weights will lead to small differences in the development of cells within each cluster, these differences are not of sufficient importance at the *functional* level to demand full simulation of each cluster separately. By contrast, the $\mathcal{F}_1$ field must contain multiple cells for each possible input item, since these cells emphatically do not activate identically in response to events, as is illustrated by the requirement that sequences such as AAAA must be able to be represented at $\mathcal{F}_1$. In all the simulations presented here, the number of $\mathcal{F}_1$ cells that can respond to a given event is assumed to be equal to the transient memory span of the working memory i.e. 4 — this is equivalent to the number of competitive clusters (notionally) present at each $\mathcal{F}_2$ field and is also assumed to be the number of weighted connections (links) that connect each $\mathcal{F}_1$ cell to each $\mathcal{F}_2$ cell. $\mathcal{F}_1$ cells that respond to the same item (albeit at different times) will henceforth

be referred to as each other's *siblings*. Similarly, $\mathcal{F}_2$ cells, in separate clusters, that respond to the same input pattern, will also be referred to as each other's siblings.

The matrix of weighted connections between $\mathcal{F}_1$ and $\mathcal{F}_2$ also contains some redundancy. For definiteness, consider the changes that must be made to a network with 7 $\mathcal{F}_1$ cells and 60 $\mathcal{F}_2$ cells, such as that used in the previous chapter, in order to make it suitable for the processing of repeated items. As noted above, the number of $\mathcal{F}_2$ cells can be left unchanged, bearing in mind that their activities summarize the activities in four separate competitive clusters. The number of $\mathcal{F}_1$ cells must increase fourfold to 28. The number of weights connecting $\mathcal{F}_1$ to a single $\mathcal{F}_2$ cluster must, therefore, increase sixteenfold — there are four times as many $\mathcal{F}_1$ cells and each pair of cells is connected by four weighted links. The prospect of having to calculate weight values $(z_{ji}, w_{ji}, z_{ij}^f)$ at sixteen times as many "sites" as before, is not a appealing one in computational terms. Fortunately, we can exploit the redundancy in the weights that results from the constraint that learning at a given $\mathcal{F}_1 - \mathcal{F}_2$ link must be identically carried out at *corresponding* links (reminder: a corresponding link is one which connects to the same $\mathcal{F}_2$ cell from a sibling $\mathcal{F}_1$ cell, and which "codes" for the item in the same list position). This constraint ensures that for each link there will be three other links with identical weight (once again ignoring small variations due to random initialization), which in turn means that the increase in the number of *different* weights, necessitated by the processing of repeated items, is only a fourfold one.

Whereas we formerly referred to a given weight with the notation z_{ji}, where j indexed the $\mathcal{F}_1$ cell and i indexed the $\mathcal{F}_2$ cell, to reflect the fourfold increase in the number of effective weights between fields, we shall now describe a weight by the notation z_{jik}, where i indexes the $\mathcal{F}_2$ cell as before, where j indexes the *item* to which the corresponding $\mathcal{F}_1$ cell responds, and k $(0 \leq k \leq 3)$ indexes the four links that connect the two cells. The fact that j indexes the *item* to which the corresponding $\mathcal{F}_1$ cell responds (remember there are still only seven possible items that can occur at this level), rather than indexing the $\mathcal{F}_1$ cell itself (of which there are 28), illustrates the point at which the redundancy described above is exploited. A similar notation is adopted for the weights formerly notated w_{ji} and z_{ij}^f. Note that there must also be multiple feedback links from a given $\mathcal{F}_2$ cell to a given $\mathcal{F}_1$ cell — the structure, and the reasoning underlying it, is almost entirely analogous to that of the bottom-up weights.

The next design decision was made with regard to the implementation of the inter-link

inhibitory connections. Whilst it would be possible to simulate these by introducing new variables representing relay-cell activations, the fact that the state of a given link is essentially binary (i.e. a link is either fully active or fully inactive) suggests a more computationally appealing solution. We introduce the concept of an $\mathcal{F}_1$ cell's *rank*, denoted R_j. The rank of an $\mathcal{F}_1$ cell is defined as the number of its siblings whose activation exceeds that of the cell itself at a given moment. If two siblings have the same activation (necessarily zero in the case of sequence processing) then the tie is broken arbitrarily. Note that the rank of a cell does not correspond to any physical property of the cell, but merely aids in the simulation of the *function* of the inter-link inhibitory connections. In this case the rank of an $\mathcal{F}_1$ cell is an integer between zero and three inclusive.

We also make two further assumptions to aid simulation: firstly, that $\mathcal{F}_1$ is arranged so that the cells corresponding to the indices j such that $0 \leq j \leq 6$ all respond to a different item, as do the cells indexed with $7 \leq j \leq 13$ etc., and, further, that cells whose indices differ by ± 7 respond to the same item; secondly, that for any given choice of j and i, z_{jik} is monotone decreasing for an increase in index k, i.e. $z_{ji0} > z_{ji1} > z_{ji2}$ etc.. If we insist that this latter constraint holds immediately after random initialization of these weights, then it can be shown that the relationship is maintained throughout weight-learning as long as a fairly liberal constraint is placed on the values of ϵ_1, ϵ_2 and the simulation time-step dt (in fact we need $\epsilon_1 \epsilon_2 dt < \frac{1}{B_2}$). The value R_j is used in the following way: whenever we use a weight value in a calculation, for instance in the calculation of I_i^+ or in the weight-learning equations, we use the weight value that is indexed $z_{\alpha i R_j}$, where $\alpha = j \bmod 7$ and j and i index the cells in $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively, as before. In other words, the most active of a set of siblings, i.e. the one with rank zero, will have its input to a given $\mathcal{F}_2$ cell gated by the largest of the four weighted links that connect the two cells; the next most active sibling will have its activity gated by the second largest weighted link and so on. This fully simulates the action of the inter-link competition, whilst remaining computationally efficient — the only additional requirement is that the rank values are calculated at each time step.

There is a further issue regarding what happens when a pattern at $\mathcal{F}_1$ is "chunked out" by the reset of *one* of the cells that represents this pattern. Using Nigrin's original idea, namely that $\mathcal{F}_1$ cells would be reset when their top-down input dropped rapidly, we would have to conclude that *all four* of the $\mathcal{F}_2$ cells that represent the pattern are reset simultaneously, since

they will all supply top-down feedback signals to the cells in their receptive field (incidentally this is another good reason to employ a "max" function, rather than a sum, in the calculation of total feedback to an $\mathcal{F}_1$ cell). Since the mechanism by which the reset of the $\mathcal{F}_2$ cell is achieved is assumed to involve inhibition by a *specific* s-cell in the layer above, Nigrin's idea seems implausible. We can conclude that direct resetting of the relevant $\mathcal{F}_1$ cells on chunking-out is necessary, as was suggested in my previous chapter. This raises the question of what should happen to the activity of its three sibling $\mathcal{F}_2$ cells, when a given $\mathcal{F}_2$ cell resets. Since the activity of the cells in their receptive field has been reset, the activity of the $\mathcal{F}_2$ cells will decay after chunking-out. Simulations have shown, however, that allowing their activity to decay in this way can interfere with the processing of subsequent items — it is preferable to assume that the activity of these cells is reset quickly. This can be achieved if we assume that an $\mathcal{F}_2$ cell resets itself when its *bottom-up* signal drops abruptly (in one sense the reverse of Nigrin's idea). Thus, when the cells in $\mathcal{F}_1$ are reset *directly* by the action of an $\mathcal{F}_2$ cell (chunking-out), the siblings of that $\mathcal{F}_2$ cell are also reset, as are any other $\mathcal{F}_2$ cells that previously relied on the same $\mathcal{F}_1$ cells for a significant part of their input. This mechanism has the advantage that $\mathcal{F}_2$ cells that respond to activity in a separate set of $\mathcal{F}_1$ cells, are unaffected by the chunking-out process and their activity can persist. It also suggests a means of implementing the $\mathcal{F}_2$ reset event that occurs when $\mathcal{F}_1$ overloads.

7.2.3 Results

As in the previous chapter, the performance of the network, now adapted to be able to handle repeated items, was assessed with regard to its ability to learn the set of melodic phrases derived from the 12 simple melodies. The original (unaltered) phrases, of which there were 29 (7 three-note, 22 four-note), can be used. The network, to which these phrases were exposed, consisted of 28 $\mathcal{F}_1$ cells, arranged into four clusters of 7 cells each, and 65 $\mathcal{F}_2$ cells. Such a network choice of network size requires quite tight "compression", the ratio of $\mathcal{F}_2$ cells to required phrase categories being only slightly greater than two. It is desirable to have tight compression, since, as noted earlier, the run-time of network simulations increases approximately with the square of the number of $\mathcal{F}_2$ cells — this constraint would obviously not apply to a parallel implementation, such as is assumed to exist *in vivo*. There was one slight difference between the simulations presented here, and those presented in Chapter 6,

namely that each sequence was held in working memory only for as long as was possible before $\mathcal{F}_1$ overload, and hence reset, occurred. Using the parameters specified below, this time was a little longer than that required to present six items.

The individual phrases were presented to the network in an incremental-learning regime, once again for four different orderings of the phrase-set. In each case, a distinct $\mathcal{F}_2$ cell became committed to each of the input phrases, and no other $\mathcal{F}_2$ cell became spuriously committed. This constitutes ideal learning of the input set. Table 7.1 shows the number of presentations required to learn each of the input phrases. It should be noted that, for the third random ordering, three of the phrases were presented more than 10 times before being learned. The numbers of presentations required are given in the form "$10 + x$" for the following reason: the initial phase of incremental training consisted of the presentation of each phrase a maximum of 10 times; subsequently any phrases that had not been learned were presented to the network until learning was achieved. This procedure was adopted for practical reasons—it allowed the computer simulations to proceed without continuous monitoring. For this reason, the number of presentations required to learn these three phrases should be considered to be "on the high side" of what might be expected for genuine incremental learning. Nonetheless, the fact that these three "outliers" all occurred on the same run, suggested a common cause — inspection of the state of the network showed that learning had been slowed by the presence of an uncommitted cell, with large weights to a small set of $\mathcal{F}_1$ items, but with its own learning slowed due to its low M-value. The interference of such a cell can clearly be overcome, as it was in each of the cases above — further work is necessary to find a way of preventing such a situation from occurring in the first place.

A single run was performed to illustrate that ideal learning of the input set could be accomplished by cyclic presentation of the input set, once again in a random ordering. The number of presentations required to learn the input phrases is shown in Table 7.2. The performance is broadly similar to that seen in the equivalent simulations in the previous chapter — the main difference is that the shortening of the phrase presentation time necessitated a larger number of cycles before learning of all patterns was achieved. Cyclic presentation was not performed for multiple random input-orderings because of the prohibitive run-time that such a regime entails.

The presentation of individual phrases, as opposed to the presentation of continuous se-

Order	Phrase	NP	Phrase	NP	Phrase	NP	Phrase	NP
1	3543	4	5342	4	1715	4	671	4
2	6231	3	543	2	671	3	3421	5
3	4321	4	151	2	2275	3	5451	3
4	4231	3	345	2	3421	2	6543	2
5	251	2	3543	2	1355	3	5342	2
6	3431	2	4231	3	5342	3	4231	3
7	671	4	671	2	1321	(10+6)	1355	2
8	1565	3	135	2	251	2	1324	2
9	345	2	2451	3	345	3	345	2
10	5432	3	5531	2	3543	2	1715	3
11	1765	2	5432	3	1231	2	321	2
12	1355	2	4321	3	1531	3	5432	2
13	1321	2	1765	3	5432	2	1151	2
14	5342	2	321	2	3431	(10+8)	543	2
15	1531	3	1531	3	4321	6	3431	2
16	5531	2	6231	3	2451	3	2451	3
17	3421	2	3421	4	6231	2	135	4
18	5451	3	1321	3	135	3	1765	4
19	1231	2	6543	4	5531	2	6231	4
20	1151	3	1324	3	5451	3	3543	3
21	2451	3	5451	4	1324	3	251	2
22	151	8	251	2	1765	3	1531	3
23	2275	3	2275	4	6543	2	1565	2
24	1715	2	1565	3	543	2	4321	3
25	1324	2	1355	3	151	2	1231	4
26	543	2	1151	2	321	(10+3)	1321	2
27	321	3	1715	3	4231	4	5531	3
28	6543	8	3431	2	1151	2	151	2
29	135	3	1231	3	1565	3	2275	3

Table 7.1: Number of presentations (NP) needed to learn each phrase for four different random orderings (incremental learning)

Order	Phrase	NP	Order	Phrase	NP
1	251	47	16	1355	9
2	1715	12	17	6231	4
3	543	20	18	151	37
4	135	32	19	3431	8
5	5451	34	20	1765	12
6	5342	17	21	5531	8
7	2275	3	22	1231	18
8	1151	2	23	3543	26
9	4231	2	24	1565	6
10	5432	8	25	1321	1
11	1531	9	26	345	83
12	671	79	27	4321	3
13	1324	2	28	6543	17
14	321	65	29	2451	15
15	3421	1			

Table 7.2: Number of presentations (NP) needed to learn each phrase with cyclic learning

quences (i.e. the melodies themselves), was another restriction imposed by the "banana" problem. Having solved this problem, it was possible to test the network's ability to learn the input phrases from presentation of the melodies themselves. This learning regime was in a sense intermediate between the incremental and cyclic presentation regimes described above: we now present melodies, rather than phrases, incrementally, until their constituent phrases are learned. Thus the network will, at a given time, be attempting to learn a small number of new phrases, rather than the single new phrase of the incremental regime, or the multiple new phrases of the cyclic presentation regime. Before discussing the results of such simulations, I will describe the details of the network required to perform phrase learning from "continuous" item-presentation.

The continuous presentation of items necessitates specification of the conditions for $\mathcal{F}_1$ overload and the resulting $\mathcal{F}_1$ reset. Leaving the $\mathcal{F}_1$ parameters unchanged (remember these were set so as to produce a "shunting" parameter approximately equal to two), these conditions can be specified by the setting of the parameters K_{13} and K_{14}. Briefly, K_{13} is set so that the field $\mathcal{F}_1$ overloads during presentation of the seventh item in a sequence presented isochronously. Since the total activation in $\mathcal{F}_1$ approximately doubles with the occurrence of each item (in fact, with the occurrence of each attentional beat), considerable latitude is available in the setting of K_{13} — in the simulations described here, $K_{13} = 10$. When overload occurs, all cells with an activation greater than K_{14} are reset. K_{14} was set so that, on overload, the first four items of an isochronously-presented sequence would be "deleted" from working memory — $K_{14} = 0.5$ was suitable. Thus, for these settings, the presentation of the sequence $(1, 2, 3, 4, 5, 6, 7)$ leads to an $\mathcal{F}_1$ overload reset, after which the pattern $(5, 6, 7)$ remains in working memory. The presentation of the sequence $(1, 2, 3^2, 4, 5, 6)$ will likewise lead to an $\mathcal{F}_1$ overload reset, but, because of the lengthening of the third item, this will only result in the deletion of the first three items in working memory, leaving the pattern $(4, 5, 6)$. This behaviour is fortuitous, given the make-up of the current input set, since it means that an $\mathcal{F}_1$ reset event will always remove an entire phrase from working memory, whether it be a four-note phrase, or a three-note phrase with a lengthened final note. This is achieved without any recourse to the additional "accent" information that was employed in Chapter 5, even though that "accent" information was held to be available to the listener by reference to alternative aspects of a melody's structure, such as the rhythmic and, in the case of rhymes, semantic

structure. It should be noted that, in general, we may not be able to rely on judicious setting of fixed $\mathcal{F}_1$ reset parameters to "discourage" the learning of 'across-phrase" groupings from continuous presentations, and that the alternative "clues" to the likely location of phrase boundaries might thus have to be invoked. This is further discussed later in the chapter.

Additionally, we must specify the conditions for the chunking-out of a learned sequence. This is deemed to occur once the activity of the corresponding $\mathcal{F}_2$ cell has continuously exceeded K_{11} for a time K_{12}. A typical parameter setting would be $K_{11} = 1.0$, whilst K_{12} would be set equivalent to the time taken to present 1–2 items. The delay in chunking-out, K_{12}, enables patterns representing supersets of the "to-be-chunked" phrase, to be learned if appropriate. This is particularly necessary in situations where, for instance, the sequence $(1, 2, 3, 4)$, is presented repeatedly, after the sequence $(1, 2, 3)$ has already been learned — if the latter pattern were to be chunked out immediately after the activation of the corresponding c-cell exceeded K_{11}, then the longer sequence would never be learned. This example illustrates the additional need to allow K_{12} to vary, depending on the size of the active c-cell. For instance, in the situation described above we would want K_{12} to be equivalent to the time taken to present at least 2–3 items, allowing the presentation of the item "4" and an additional period in which to learn the longer sequence. If, however, the sequence $(1, 2)$ has previously been learned, its chunking-out would need to be delayed for a period equivalent to the time taken to present 3–4 items, if the longer sequence $(1, 2, 3, 4)$ is to be learned. For the current simulations K_{12} was set to 0.3 for size-4 cells (i.e. cells corresponding to four-note sequences), 0.5 for size-3 cells, 0.7 for size-2 cells etc.. This is equivalent to restricting chunking-out to situations in which 5–6 items are active in working memory. There are circumstances in which the chunking-out of a pattern should not be delayed in this way, namely in situations where "hierarchical" expectations are present — this is discussed later.

These changes having been made, the network is now able to learn phrases from exposure to whole melodies. Unlike with **MASKART**, in which the high-vigilance regime led to problems, the presence of "extraneous" items in working memory does not affect the learning of a given phrase, provided the ordering and timing of items is correct. The upper limit on the length of learned phrases is effectively set by the level of K_4. As in the previous chapter, this was set to $K_4 = 12$, which allows phrases of up to four items to be learned.

To test the behaviour of the network, each melody was presented repeatedly to a network

Melody	Phrase	NP	Phrase	NP	Phrase	NP	Phrase	NP
1	2451	2	5451	4	—	—	—	—
2	1324	2	3421	2	—	—	—	—
3	321	2	543	2	1715	2	—	—
4	4321	2	5432	4	1565	4	—	—
5	2275	2	251	2	1151	4	—	—
6	1231	2	6231	3	—	—	—	—
7	4231	2	1355	4	—	—	—	—
8	2275	2	3431	2	671	3	5531	4
9	1565	3	1531	3	4321	4	1765	4
10	6543	3	3543	4	1321	4	—	—
11	5342	2	151	2	135	4	—	—
12	1231	2	345	2	5531	2	151	2

Table 7.3: Number of presentations (NP) of each melody needed to learn each constituent phrase

at which no previous learning had been carried out. The network had 28 $\mathcal{F}_1$ cells and 20 $\mathcal{F}_2$ cells, and the learning-rate, ϵ_2 was set to 6.0. Table 7.3 shows the number of presentations of each melody required to permit learning of each of its constituent phrases. Once again ideal learning is accomplished for each melody.

A single network consisting of 28 $\mathcal{F}_1$ cells and 65 $\mathcal{F}_2$ cells was then exposed incrementally to all of the melodies, i.e. a melody was presented sufficiently often for all of its constituent phrases to be learned, before the next melody was presented. The order in which the melodies were presented was random and the performance of the network is shown in Table 7.4. This describes the number of times each melody was presented before each of its constituent phrases were learned. Once again ideal learning was achieved, i.e. all input phrases, and no spurious phrases, were learned.

There are several points to note about this incremental melody-learning. Firstly, a large number of presentations of Melody 11 were needed before the phrase $(1, 3, 5^2)$ could be learned. This can be accounted for as follows: in Melody 11, the sequence $(1, 3, 5^2)$ is succeeded by the

Melody	Phrase	NP	Phrase	NP	Phrase	NP	Phrase	NP
4	4321	3	5432	3	1565	3	—	—
3	1715	1	321	2	543	2	—	—
6	1231	1	6231	4	—	—	—	—
12	1231	—	345	1	5531	2	151	2
5	1151	2	2275	3	251	3	—	—
8	2275	—	5531	—	671	2	3431	3
10	1321	3	3543	6	6543	6	—	—
7	1355	2	4231	3	—	—	—	—
11	151	—	5342	3	135	21	—	—
2	1324	2	3421	1	—	—	—	—
9	1565	—	4321	—	1765	6	1531	4
1	5451	2	2451	4	—	—	—	—

Table 7.4: Number of presentations (NP) of each melody needed to learn each constituent phrase - incremental melody presentation

item "5"; once this item has been presented, the sequence $(1, 3, 5^2, 5)$ is represented in working memory. Prior presentation of Melody 7, however, will have enabled the network to learn the phrase $(1, 3, 5, 5)$, and it is $\chi_{1,3,5,5}$ which activates so as to inhibit the activation of the cell that is attempting to learn the three-note phrase. Eventually the shorter pattern is learned. The second point to be made, concerns the order in which patterns are learned. Examination of the (random) order in which the melodies were presented, shows that the network was able to learn the phrase $(5, 4, 3, 2)$ *before* being expected to learn the pattern $(5, 4, 3)$ — this is fortuitous since, had the sequence $(5, 4, 3)$ been learned first, then on presentation of Melody 1, $(5, 4, 3)$ would have been chunked out leading to the possible learning of across-phrase sequences. With the current network there seems to be no simple way to prevent such an occurrence, other than by appealing to the other grouping "clues" referred to above, which might intervene in such a way as to inhibit the inappropriate chunking-out of the three-note phrase. Further work is necessary before suggesting a mechanism by which this might be accomplished (see Chapter 9).

7.2.4 The generation of expectations

The previous sections have showed how the phrases that constitute the melody-set, can be learned under a variety of different regimes: incremental presentation of isolated phrases; cyclic presentation of isolated phrases; individual presentation of melodies; and incremental presentation of melodies. All regimes led to ideal learning. This section shows the expectations that are elicited from a network that has learned these phrases (regardless of the regime under which they were learned), and demonstrates the network's ability to expect item repetitions.

A network that had learned all 29 input phrases was presented with the relevant set of one-, two- and three-item probes, and the expectations thus elicited were recorded. These are shown in Tables 7.5-7.7. Since $\mathcal{F}_1$ now contains 28 cells, the expectations are displayed in a different manner from that used in the previous chapter. The identities of the most expected items are given, together with a number indicating the relative expectations for those items. As before, a code is supplied indicating the "validity" of the expectations elicited, with reference to the entire input-set. Using this means of assessment, all of the expectations elicited by the one-, two- and three-item probes were deemed to be valid, including those in which expectations for item repetition were generated.

Probe	Expectations						Code
1	3(0.023)	5(0.021)	1(0.010)	2(0.006)	7(0.006)	6(0.003)	*
2	5(0.023)	1(0.011)	4(0.006)	2(0.006)	7(0.003)	—	*
3	2(0.023)	4(0.021)	1(0.011)	5(0.010)	3(0.003)	—	*
4	2(0.006)	3(0.005)	1(0.001)	—	—	—	*
5	4(0.023)	3(0.011)	5(0.006)	2(0.001)	1(0.001)	—	*
6	7(0.023)	1(0.011)	5(0.006)	2(0.006)	3(0.003)	4(0.003)	*

Table 7.5:

Whilst, the expectations elicited are all valid, we still have the situation in which expectations elicited from $\mathcal{F}_2$ cells corresponding to shorter phrases dominate those from cells corresponding to longer phrases. Thus the two-item probe $(1,3)$ always elicits primary expectation for a "5", as a result of the activation of $\chi_{1,3,5}$, in spite of the fact that the probe sequence is, *in certain contexts*, always followed by a "2". In order to address this point we must provide a mechanism whereby additional context can influence expectations. The next section introduces just such a mechanism.

7.3 The provision of a hierarchy

7.3.1 Structure

The results of the previous section indicate that, in order to provide veridical memory of extended, multi-phrase sequences, we must use a hierarchy of **SONNET** modules. The basic idea is simple: once we reach the end of a melodic phrase, there is no further note expected *at that level* — to use a musical term, we have "closure" at the note level; any expectations for the next event must, therefore, draw on additional knowledge concerning the likely identity of the next *phrase*. This requires knowledge of familiar phrase-sequences, in the same way as note expectations require knowledge of familiar note-sequences. As the hierarchy increases in depth, longer sequences can be stored in memory providing that, at each level, the sequence can be divided into familiar chunks. This section shows how **SONNET** modules can be reasonably easily arranged in a hierarchical fashion and how adjacent levels should interact

Probe	Expectations					Code
1,1	5(0.014)	1(0.006)	—	—	—	*
1,2	3(0.013)	1(0.006)	—	—	—	*
1,3	5(0.054)	2(0.015)	1(0.007)	5(0.006)	4(0.006)	*
1,5	1(0.051)	3(0.015)	6(0.015)	5(0.007)	—	*
1,7	1(0.014)	6(0.014)	5(0.006)	—	—	*
2,2	7(0.014)	5(0.006)	—	—	—	*
2,4	5(0.014)	1(0.007)	—	—	—	*
2,5	1(0.054)	—	—	—	—	*
3,2	1(0.054)	—	—	—	—	*
3,4	5(0.050)	2(0.014)	3(0.014)	1(0.007)	—	*
3,5	4(0.015)	3(0.007)	—	—	—	*
4,2	3(0.014)	1(0.006)	—	—	—	*
4,3	2(0.014)	1(0.007)	—	—	—	*
5,3	4(0.015)	2(0.007)	—	—	—	*
5,4	3(0.055)	5(0.014)	1(0.007)	2(0.006)	—	*
5,5	3(0.015)	1(0.007)	—	—	—	*
6,2	3(0.014)	1(0.007)	—	—	—	*
6,5	4(0.014)	3(0.007)	—	—	—	*
6,7	1(0.054)	—	—	—	—	*

Table 7.6:

Probe	Expectations		Code
1,1,5	1(0.025)	—	*
1,2,3	1(0.019)	—	*
1,3,2	1(0.025)	4(0.023)	*
1,3,5	5(0.018)	—	*
1,5,3	1(0.023)	—	*
1,5,6	5(0.026)	—	*
1,7,1	5(0.024)	—	*
1,7,6	5(0.024)	—	*
2,2,7	5(0.025)	—	*
2,4,5	1(0.027)	—	*
3,4,2	1(0.026)	—	*
3,4,3	1(0.025)	—	*
3,5,4	3(0.026)	—	*
4,2,3	1(0.025)	—	*
4,3,2	1(0.025)	—	*
5,3,4	2(0.026)	—	*
5,4,3	2(0.017)	—	*
5,4,5	1(0.025)	—	*
5,5,3	1(0.027)	—	*
6,2,3	1(0.026)	—	*
6,5,4	3(0.026)	—	*

Table 7.7:

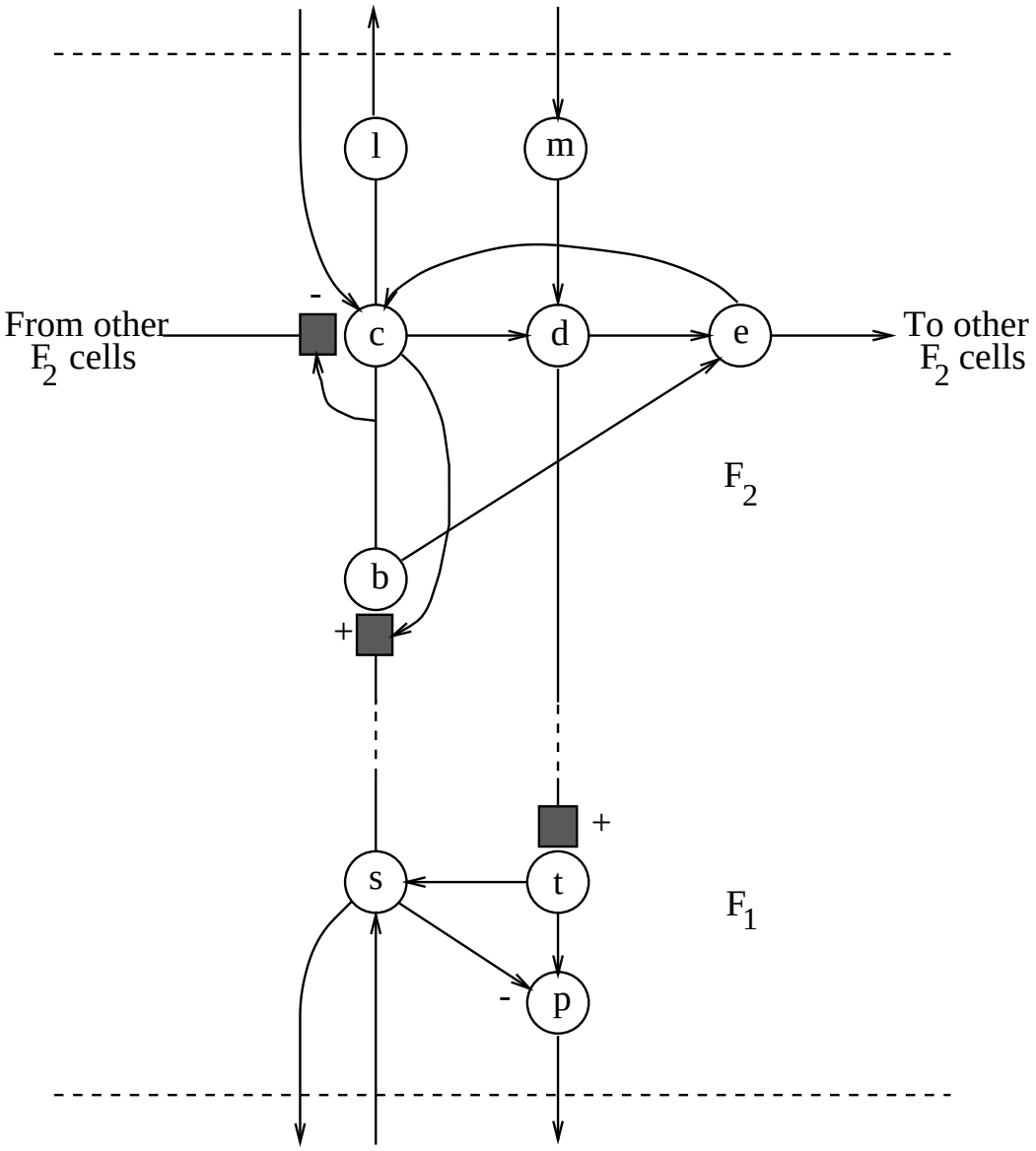


Figure 7.3: The connectivity within a single module.

to allow the learning of multi-phrase sequences.

Most of the structure necessary for the building of **SONNET** hierarchies, has already been mentioned during the first part of this chapter. The connectivity within a single module is show in Figure 7.3. Each s-cell in a given $\mathcal{F}_1$ field receives bottom-up input from an l-cell (latch cell) which activates, and remains active for a short time, either in response to an environmental stimulus, such as the occurrence of a note-event, or in response to the classification of a pattern by a c-cell in the module below — this obviously depends on the position of the module in the hierarchy. As is noted in the previous chapter, and expanded upon in the next chapter, changes in activation and weights within a given module are only enabled for a certain time

period after the occurrence, or the expected occurrence (see Section 6.3), of an event at the input of that module. This is to allow the same operational parameters to be used for all modules, regardless of their position (height) in the hierarchy. Much of the advantage to be gained from the building of a hierarchy concerns the generation of top-down signals which can lead to "hierarchical expectations". As can be seen in Figure 7.3, top-down feedback (from cell assemblies that meet the relevant conditions) is mediated by the d-cell, which itself receives input from two sources: firstly, it receives input from the c-cell, so that those c-cells that are able to compete best for activation in response to a given $\mathcal{F}_1$ pattern, are accordingly able to contribute most to the top-down feedback; secondly, it receives input from the t-cell in the module above — this is absolutely essential to allow priming to percolate down through the hierarchy. To illustrate why this is so important, imagine that the network has learned the first two phrases of a melody, e.g. $(1,2,3,4)(5,6,7^2)$: after the "4" has been presented there will be no expectation generated from within the note-level module, since the phrase will be deemed complete; neither will there be any bottom-up input to the $\mathcal{F}_2$ cell $\chi_{5,6,7}$, which in turn implies that $c_{5,6,7}$ will be zero. In order, therefore, for there to be any expectation for the next note, the "5", $d_{5,6,7}$ will require activation from a source other than its associated c-cell. This activation can come from the module above, providing that the cell that corresponds to the entire phrase $(5,6,7)$ is itself expected at that level. This will be the case if the *phrase-sequence* $(1,2,3,4)(5,6,7)$ is itself familiar. Before testing the ability of this mechanism to function as desired, in the learning of extended sequences, I shall describe its implementation.

7.3.2 Implementation

The program to simulate the behaviour of a **SONNET** network extended in the ways indicated above, was written with many of the same constraints as were applied to the masking field, **MASKART** and **SONNET 1** simulations. Realistic experimentation requires that the user has a means of assessing the state of the network at a given moment. For each $\mathcal{F}_1 - \mathcal{F}_2$ module, there is an animated graphical display of the (normalized) $\mathcal{F}_1$ activations, s_j, the $\mathcal{F}_2$ activations, c_i, the top-down feedback signals from $\mathcal{F}_2$ to $\mathcal{F}_1$ and the resultant expectations. The expectation vector is normalized for display purposes, and an adjacent number indicates the strength of expectation for the most expected item. The display of the $\mathcal{F}_1$ activations is grouped into four sections, corresponding to the four clusters, each of which contains a cell

for each of the possible input items. Each $\mathcal{F}_1$-$\mathcal{F}_2$ module has a "Modify Parameters" button which pops-up a window containing the current parameter settings for that module — in contrast with the **SONNET 1** simulations, a pop-up window is used to save on "screen real-estate", which is at a premium when displaying hierarchies. Parameter settings can be edited by the user whilst the program is running. The ability to edit parameters at run-time also extends to numbers of cells in each field, so memory for the module variables must be dynamically allocated and deallocated when appropriate. In addition, whenever the field sizes are changed, the program automatically checks that they are consistent for modules adjacent in the hierarchy. Each module also contains buttons for loading and saving that module's weight matrices etc., via a series of pop-up file browsers. The graphical display can be disabled to allow accelerated running and modules can be selectively disabled at run time.

The most important aspect of the program's design concerns the use of object-oriented programming techniques. The fact that a hierarchy consists of a series of functionally identical modules, strongly suggests an approach that takes advantage of such duplication — object orientation is just such an approach. Briefly, an *object* consists of a block of data together with the set of procedures (also referred to as "methods") that are permitted to act on that data. This tight association of the data with the methods that are permitted to access and modify them, is termed *encapsulation*. Objects are "instances" of a *class* — the class of an object is an abstraction, which describes the structure of the data and *defines* the associated methods. An object-oriented approach also embraces concepts such as the *inheritance* of data structures and methods, though such concepts are not as relevant to the current application.

In the case of the extended **SONNET** program, object-orientation was applied at a relatively high level: a class was defined to correspond with the structure and behaviour of a single $\mathcal{F}_1$-$\mathcal{F}_2$ module. This class defines all the variables associated with a given module, i.e. cell activations, weights, parameters etc., and all the methods that describe the ways in which the variables interact over time. The encapsulation of the data and methods within an $\mathcal{F}_1$-$\mathcal{F}_2$ *object* mirrors the "local" nature of the interactions within modules, in the sense that changes in the variables in a given module should depend on information available within that module. There are exceptions to this general rule, i.e processes involving the passing of information between modules. The extents of a module are shown in Figure 7.3 — accordingly, a given module must have access to the state of the latch cells in the module below, and the s- and

t-cells in the module above. This minimal interaction between modules is easily incorporated into the object-oriented design.

The attentional signal which enables changes in activity at a given module, is also easily modelled. Each module keeps track of the time since it last received an input event; if this time exceeds a certain value then processing is disabled at that module. This strategy also minimizes the computational penalty of building hierarchies: at each time step, all modules in a hierarchy are sent a "message" inviting them to advance their state by a single time step. Modules that are disabled simply return the message without performing any processing, i.e. without incurring any computational cost. If we have a two-module hierarchy, we can expect that the higher module will only be enabled for approximately a third of the time that the lower module. The addition of the higher-module, therefore, only requires an increase in computation of about 33% (assuming the modules have the same numbers of $\mathcal{F}_1$ and $\mathcal{F}_2$ cells). The addition of a third module in the hierarchy only incurs an increase in computational load of about 45%, compared with the single module case. In the current application, the *rhythmic* attentional signal was only simulated at the note level.

The object-oriented approach makes the building of hierarchies extremely easy: the addition of a module merely involves instantiating a new module object and making a small number of "message-passing" connections to the existing structure — this can be done graphically. Finally, at the "root" of each hierarchy is an "event module", which is simply an array of latch cells which respond to environmental events, rather than to chunking at a lower module. A screen dump showing the appearance of the program's interface is shown in Figure 7.4.

7.3.3 Learning phrases of phrases

Once the network is structured in the way described above, the learning of phrase sequences is essentially no different from the learning of note sequences at the lower level. Each $\mathcal{F}_1$–$\mathcal{F}_2$ module sees only the arrival of "items" at $\mathcal{F}_1$, and is not concerned whether these items represent notes, phrases, chords, letters, phonemes etc.. Thus repetition of a sequence of, say, four *familiar* phrases at the lower level, will lead to the learning of the four-item sequence that results at the higher level, once all phrases have been chunked out. Note that the phrase sequence can only be learned at the higher-level once each of the phrases is sufficiently

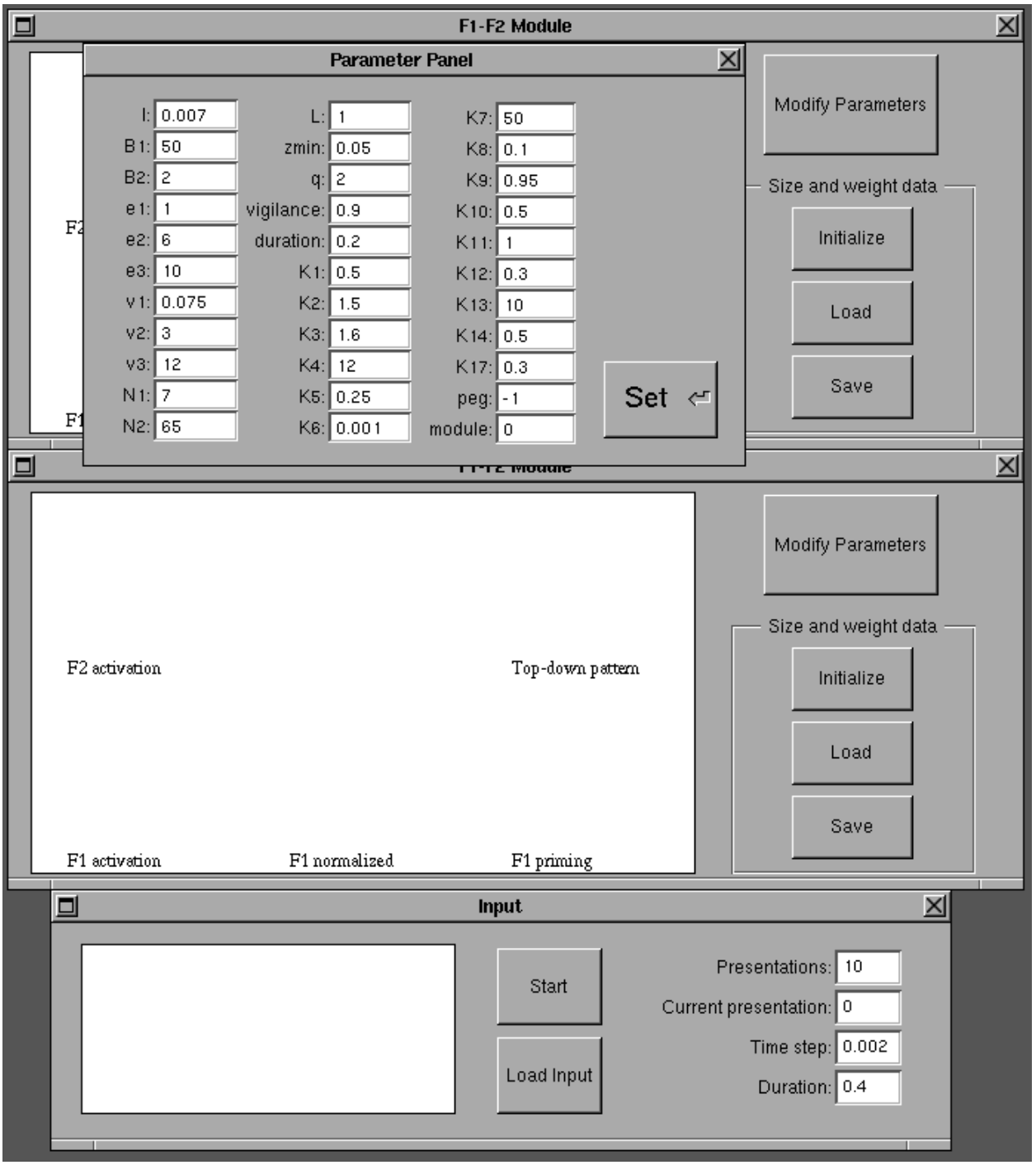


Figure 7.4: A screen dump of the extended-**SONNET** simulation

familiar at the lower level to be represented by a committed $\mathcal{F}_2$ cell. In the learning of, say, a melody from repeated exposure, we thus must assume that significant learning of higher-order structure succeeds the learning of the lower-order structure, i.e. the learning of the phrases themselves. To give a linguistic analogy, we cannot expect that a sentence is learned until all of the constituent words are themselves learned. Practically, we can ensure that this is the case in a number of ways. The first possibility is that each of the musical phrases is learned in isolation before proceeding on to the learning of the entire melody — this is reminiscent of the way in which choral societies approach the learning of a new piece (for the benefit of those not sufficiently skilled to read the music!): short sections are learned before being strung together to form the whole. Secondly, we can make sure that the (fixed) learning rate at the higher-level module is set relative to that at the lower-level module in such a way as to ensure that the note-phrases are learned faster than the phrase-phrases. Thirdly, we can postulate that learning is inhibited at higher-level modules at times when $\mathcal{F}_1$ overloads are commonly occurring in lower-level modules. The current simulations side-stepped this issue by ensuring that all note-phrases were learned (in any one of the regimes described above) before learning at the higher-level was enabled. Once the note-phrases were learned, repeated presentation of the melodies led to the learning of phrase-phrases as expected.

One important point emerged during simulations, regarding the network's ability to learn phrase-phrases. The higher-level module had 65 $\mathcal{F}_1$ cells and 30 $\mathcal{F}_2$ cells, and was set the task of learning the phrase-sequences corresponding to the 12 melodies. Eight of the melodies consisted of only four phrases, and could therefore be represented by a single phrase-phrase. The remaining melodies could only be represented by two phrase-phrases, and strictly speaking would therefore require a third module to learn the single "phrase-phrase-phrase" corresponding to the melody. This third module was not simulated since its presence is irrelevant to the ideas behind the building of hierarchies. The learning of phrase-phrases at the relevant module became very slow, however, once several melodies had already been learned. It was realized that this slowing was caused by the fact that the remaining *uncommitted* cells at $\mathcal{F}_2$ had become *partially* committed to earlier patterns, and that these patterns had *no overlap* with the patterns subsequently to be learned. Thus presentation of the latter patterns led to very little activation of uncommitted $\mathcal{F}_2$ cells, and learning was drastically slowed. This problem was obscured during the learning of note-phrases, and presumably during the simula-

tions presented in [117], since the restricted set of 7 possible input items ensured at least some overlap between patterns presented later and some of those which had been learned earlier. The problem emerges partly as a result of the desire, for practical purposes, to limit the size of $\mathcal{F}_2$, and would disappear if this constraint were relaxed. Nonetheless, a solution was adopted which postulated that the activation of a b-cell at an uncommitted cell assembly, should not drop below a given minimum value in the region of 0.05 — ideally a random element is incorporated into the setting of this value at a given uncommitted cell to avoid identical learning at several uncommitted cells in response to a novel pattern. This solution worked well.

7.3.4 Hierarchical expectations

Having supplied the network structure necessary to permit the formation of higher-level categories (phrases of phrases), and having learned such categories on exposure to the melody-set, we now investigate the means by which higher-level knowledge can be reflected in the expectations evident at the lower (note) level. These expectations must permit, when appropriate, the recall of any of the learned melodies, i.e. they must, given sufficient context, be veridical expectations.

Firstly, we must establish the manner in which a higher-level module should communicate its phrase expectations to the lower level. Clearly, a higher-level module should convey to a lower-level module the identity of phrases which are expected to occur *next*, but not the identity of phrases which are expected to occur only sometime in the future. The distinction is important: in response to a given initial phrase, the higher-level module might produce a set of graded expectations for the set of phrases to come; we only really want to prime for the occurrence of the phrase expected to occur immediately next. Thus we postulate an inter-**m**odule priming field, with cells labelled m_i, which number the same as the $\mathcal{F}_1$ cells in the field above. The activations m_i are set by

$$m_i = \begin{cases} 1.0 & \text{if } p_i > 0.85 p_{max} \text{ and } p_{max} > 0 \\ 0.0 & \text{otherwise} \end{cases} \tag{7.1}$$

where p_i refers to the expectations in the module above, and p_{max} refers to the maximum value of p_i found in that module. The value of 0.85 is chosen to allow "graded" expectations to be excluded, whilst genuine ambiguity is reflected.

We have seen how the inter-module priming field is influenced by the module above. Now

we must specify how it influences the module below. This influence should achieve two things: firstly, it should facilitate the chunking-out of expected phrases relative to unexpected phrases; secondly, it should allow the generation of veridical expectations at the lower-level. The latter aim is achieved by allowing activation at an inter-module m-cell to affect the corresponding d-cell at the lower level so that the activations d_i are now given by

$$d_i = \min((c_i + K_{17} m_i), 1.0) \tag{7.2}$$

where K_{17} is set with the constraint that it must be large enough to have the desired effect on the expectations at the lower level. To illustrate this point, suppose the phrases $(1, 1, 2, 2)$, $(3, 4)$ and $(3, 5, 6, 7)$ have been learned at a low-level module and further that the phrase-phrase $((1, 1, 2, 2), (3, 5, 6, 7))$ has been learned at the module immediately above. On presentation of the sequence $(1, 1, 2, 2, 3)$ we would therefore want the pattern $(1, 1, 2, 2)$ to be chunked out, leaving item "3" in working memory; given the expectation that the phrase $(3, 5, 6, 7)$ follows the phrase $(1, 1, 2, 2)$, we would want the "3" to generate a greater expectation for the "5" than for the "4", other things being equal — this is the reverse of what we would get in the absence of phrase expectations. Simulations have shown that setting $K_{17} \approx 0.3$ gives the desired behaviour in this and similar situations. It should be noted that, as before, only on-target cells are able to generate top-down signals, regardless of whether or not they are primed from above. Recall that if there are no items active at $\mathcal{F}_1$, then all committed $\mathcal{F}_2$ cells are on-target.

The expectations can only evolve as described above once the first phrase, $(1, 1, 2, 2)$, has been chunked out, since only then will the phrase-expectation for the phrase $(3, 5, 6, 7)$ have any effect at the lower level. In fact, in normal circumstances, the chunking-out of $(1, 1, 2, 2)$ is necessarily delayed by a time K_{12} as described above, and until it is chunked out expectations at the lower module will be zero. Thus the delay in chunking-out has the consequence that phrase-initiating notes are never expected. To address this problem, it is postulated that phrases that are expected at a higher level can be chunked out almost immediately after they are "complete" at the lower level. This is achieved in two ways: firstly, top-down input from an inter-module cell, m_i, is able to enhance the corresponding value of b_i, so that $\mathcal{F}_2$ cells corresponding to expected phrases enjoy a competitive advantage; secondly, $\mathcal{F}_2$ cells corresponding to expected phrases are not subjected to the chunking-out delay K_{12}, though for chunking-out to occur we do require the M-value for the relevant cell to be close to unity

— this prevents chunking-out before the completion of the phrase. The activations b_i are thus given by

$$b_i = \begin{cases} 2I_i^+ I_i^\times & \text{if } m_i = 1.0 \\ I_i^+ I_i^\times & \text{otherwise} \end{cases} \tag{7.3}$$

This change enables primed cells to reach the threshold K_{11}, whereupon chunking-out occurs immediately, before the arrival of the next item.

The only remaining problem concerns the chunking-out of the first phrase in, say, a melody. If the identity of the melody is not known in advance, then the first phrase cannot be primed from above and therefore will be subject to a chunking-out delay. The only way in which we can expect the first phrase, and thus the first note, of a melody is by deriving this expectation from context information not itself based on the occurrence of previous notes. The most obvious example of such context information would be the identifying of a melody, either by explicit naming or by giving sufficient information in the form of, perhaps, the associated rhythm or other clues (as in the T.V. game "Name That Tune"). It is suggested that such context information can result in the "external" activation of the d-cell corresponding to the highest-level c-cell that activates in response to the relevant melody (sequence). Thus, if the *recognition* of a given melody is characterized by the activation of a given c-cell in a high-level module, and that that activation can "cue" the melody's identification, then *recall* of that melody will be accomplished by activating the associated d-cell, not from the bottom up, but "laterally".

To test this, a two-module network was fully trained with the melody-set, i.e. all phrases were learned at one module, all phrase-phrases learned at the other, and for each melody the corresponding d-cell in the upper-module was "pegged" to $d = 1.0$. The expectations for the first note were recorded. The melody was then "played" in the usual manner, and the expectations recorded following each note-occurrence. The results are shown in Tables 7.8-7.19. They show that "naming" of the melody resulted in perfect "recall" of all the learned tunes.

Thus the prior naming of a melody can enable the correct generation of expectations for all notes in the melody, including the first note. If we assume that, at any point, the network can "perform" the most expected note, then the performance of all the notes in a melody, in the correct order, can be achieved given prior identification of that melody. Thus the degree to which an expectation is veridical, as opposed to schematic, is indicated by the degree to

Melody 1: 5 4 5 1 2 4 5 1 5 4 5 1 2 4 5 1						
Working Memory	Expectations					Code
—	5(0.261)	4(0.126)	5(0.059)	1(0.028)	—	*
5	4(0.138)	5(0.065)	1(0.030)	3(0.011)	2(0.001)	*
5,4	5(0.088)	3(0.053)	1(0.041)	2(0.006)	—	*
5,4,5	1(0.080)	—	—	—	—	*
—	2(0.260)	4(0.122)	5(0.057)	1(0.027)	—	*
2	4(0.134)	5(0.063)	1(0.029)	2(0.005)	7(0.003)	*
2,4	5(0.086)	1(0.040)	—	—	—	*
2,4,5	1(0.081)	—	—	—	—	*
—	5(0.261)	4(0.126)	5(0.059)	1(0.028)	—	*
5	4(0.138)	5(0.065)	1(0.030)	3(0.011)	2(0.001)	
5,4	5(0.088)	3(0.052)	1(0.041)	2(0.006)	—	*
5,4,5	1(0.079)	—	—	—	—	*
—	2(0.262)	4(0.123)	5(0.057)	1(0.027)	—	*
2	4(0.134)	5(0.063)	1(0.029)	2(0.005)	7(0.003)	*
2,4	5(0.086)	1(0.040)	—	—	—	*
2,4,5	1(0.080)	—	—	—	—	*
—	—	—	—	—	—	*

Table 7.8:

Melody 2: 1 3 2 4 3 4 2 1 1 3 2 4 3 4 2 1						
Working Memory	Expectations					Code
—	1(0.240)	3(0.112)	2(0.052)	4(0.024)	—	*
1	3(0.122)	2(0.057)	4(0.027)	5(0.021)	1(0.010)	*
1,3	2(0.074)	5(0.042)	4(0.035)	1(0.006)	5(0.006)	*
1,3,2	4(0.068)	1(0.018)	—	—	—	*
—	3(0.262)	4(0.127)	2(0.060)	1(0.028)	—	*
3	4(0.139)	2(0.066)	1(0.031)	5(0.010)	3(0.003)	*
3,4	2(0.089)	5(0.048)	1(0.042)	3(0.014)	—	*
3,4,2	1(0.081)	—	—	—	—	*
—	1(0.249)	3(0.116)	2(0.054)	4(0.025)	—	*
1	3(0.127)	2(0.059)	4(0.027)	5(0.021)	1(0.010)	*
1,3	2(0.079)	5(0.044)	4(0.037)	1(0.006)	5(0.006)	*
1,3,2	4(0.071)	1(0.018)	—	—	—	*
—	3(0.262)	4(0.127)	2(0.060)	1(0.028)	—	*
3	4(0.139)	2(0.066)	1(0.031)	5(0.010)	3(0.003)	*
3,4	2(0.089)	5(0.048)	1(0.042)	3(0.014)	—	*
3,4,2	1(0.081)	—	—	—	—	*
—	—	—	—	—	—	

Table 7.9:

Melody 3: 3 2 1^2 3 2 1^2 5 4 3^2 5 4 3^2 $\underline{1}$ 7 $\underline{1}$ 5 $\underline{1}$ 7 $\underline{1}$ 5 $\underline{1}$ 7 $\underline{1}$ 5 3 2 1^2						
Working Memory	Expectations					Code
—	3(0.264)	2(0.127)	1(0.060)	—	—	*
3	2(0.173)	1(0.082)	4(0.021)	5(0.010)	3(0.003)	*
3,2	1(0.170)	—	—	—	—	*
—	3(0.264)	2(0.127)	1(0.060)	—	—	*
3	2(0.173)	1(0.082)	4(0.021)	5(0.010)	3(0.003)	*
3,2	1(0.170)	—	—	—	—	*
—	5(0.266)	4(0.127)	3(0.060)	—	—	*
5	4(0.174)	3(0.082)	5(0.006)	2(0.001)	1(0.001)	*
5,4	3(0.173)	5(0.012)	1(0.006)	2(0.005)	—	*
—	5(0.266)	4(0.127)	3(0.060)	6(0.000)	7(0.000)	*
5	4(0.174)	3(0.082)	5(0.006)	2(0.001)	1(0.001)	*
5,4	3(0.173)	5(0.012)	1(0.006)	2(0.005)	—	*
—	1(0.252)	7(0.122)	1(0.058)	5(0.027)	—	*
1	7(0.133)	1(0.063)	5(0.030)	3(0.023)	2(0.006)	*
1,7	1(0.086)	5(0.040)	6(0.013)	—	—	*
1,7,1	5(0.078)	—	—	—	—	*
—	1(0.252)	7(0.122)	1(0.058)	5(0.027)	—	*
1	7(0.133)	1(0.063)	5(0.030)	3(0.023)	2(0.006)	*
1,7	1(0.085)	5(0.040)	6(0.013)	—	—	*
1,7,1	5(0.073)	—	—	—	—	*
—	1(0.252)	7(0.122)	1(0.058)	5(0.027)	—	*
1	7(0.133)	1(0.063)	5(0.030)	3(0.023)	2(0.006)	*
1,7	1(0.085)	5(0.040)	6(0.013)	—	—	*
1,7,1	5(0.068)	—	—	—	—	*
—	3(0.264)	2(0.127)	1(0.060)	—	—	*
3	2(0.173)	1(0.082)	4(0.021)	5(0.010)	3(0.003)	*
3,2	1(0.170)	—	—	—	—	*
—	—	—	—	—	—	*

Table 7.10:

Melody 4: 1 5 6 5 4 3 2 1 5 4 3 2 5 4 3 2 1 5 6 5 4 3 2 1						
Working Memory	Expectations					Code
—	1(0.260)	5(0.131)	6(0.063)	5(0.030)	—	*
1	5(0.143)	6(0.069)	5(0.033)	3(0.023)	1(0.010)	*
1,5	6(0.093)	1(0.048)	5(0.044)	3(0.015)	—	*
1,5,6	5(0.082)	—	—	—	—	*
—	4(0.239)	3(0.119)	2(0.057)	1(0.027)	—	*
4	3(0.130)	2(0.063)	1(0.030)	—	—	*
4,3	2(0.086)	1(0.040)	—	—	—	*
4,3,2	1(0.078)	—	—	—	—	*
—	5(0.251)	4(0.121)	3(0.057)	2(0.027)	—	*
5	4(0.132)	3(0.062)	2(0.029)	5(0.006)	1(0.001)	*
5,4	3(0.085)	2(0.040)	5(0.014)	1(0.006)	—	*
5,4,3	2(0.069)	—	—	—	—	*
—	5(0.251)	4(0.121)	3(0.057)	2(0.027)	—	*
5	4(0.132)	3(0.062)	2(0.029)	5(0.006)	1(0.001)	*
5,4	3(0.085)	2(0.040)	5(0.014)	1(0.006)	—	*
5,4,3	2(0.069)	—	—	—	—	*
—	1(0.260)	5(0.131)	6(0.063)	5(0.030)	—	*
1	5(0.143)	6(0.069)	5(0.033)	3(0.023)	1(0.010)	*
1,5	6(0.093)	1(0.047)	5(0.044)	3(0.015)	—	*
1,5,6	5(0.081)	—	—	—	—	*
—	4(0.239)	3(0.119)	2(0.057)	1(0.027)	—	*
4	3(0.130)	2(0.063)	1(0.030)	—	—	*
4,3	2(0.086)	1(0.040)	—	—	—	*
4,3,2	1(0.078)	—	—	—	—	*
—	—	—	—	—	—	*

Table 7.11:

Melody 5: 1 1 5 1 2 2 $\underline{7}$ $\underline{5}$ 1 1 5 1 2 $\underline{5}$ 1^2						
Working Memory	Expectations					Code
—	1(0.257)	1(0.123)	5(0.058)	1(0.027)	—	*
1	1(0.135)	5(0.064)	1(0.030)	3(0.023)	2(0.006)	*
1,1	5(0.087)	1(0.041)	—	—	—	*
1,1,5	1(0.078)	—	—	—	—	*
—	2(0.252)	2(0.122)	7(0.057)	5(0.027)	—	*
2	2(0.133)	7(0.063)	5(0.029)	1(0.011)	4(0.006)	*
2,2	7(0.085)	5(0.040)	—	—	—	*
2,2,7	5(0.078)	—	—	—	—	*
—	1(0.257)	1(0.123)	5(0.058)	1(0.027)	—	*
1	1(0.135)	5(0.064)	1(0.030)	3(0.023)	2(0.006)	*
1,1	5(0.085)	1(0.040)	—	—	—	*
1,1,5	1(0.072)	—	—	—	—	*
—	2(0.264)	5(0.126)	1(0.059)	—	—	*
2	5(0.173)	1(0.081)	4(0.006)	2(0.005)	7(0.003)	*
2,5	1(0.169)	—	—	—	—	*
—	—	—	—	—	—	*

Table 7.12:

Melody 6: 1 2 3 1 1 2 3 1 1 2 3 1 6 2 3 1						
Working Memory	Expectations					Code
—	1(0.260)	2(0.125)	3(0.059)	1(0.028)	—	*
1	2(0.137)	3(0.064)	1(0.030)	5(0.021)	7(0.006)	*
1,2	3(0.088)	1(0.041)	—	—	—	*
1,2,3	1(0.080)	—	—	—	—	*
—	1(0.260)	2(0.125)	3(0.059)	1(0.028)	—	*
1	2(0.137)	3(0.064)	1(0.030)	5(0.021)	7(0.006)	*
1,2	3(0.088)	1(0.041)	—	—	—	*
1,2,3	1(0.080)	—	—	—	—	*
—	1(0.260)	2(0.125)	3(0.059)	1(0.028)	—	*
1	2(0.137)	3(0.064)	1(0.030)	5(0.021)	7(0.006)	*
1,2	3(0.088)	1(0.041)	—	—	—	*
1,2,3	1(0.079)	—	—	—	—	*
—	6(0.260)	2(0.125)	3(0.059)	1(0.028)	—	*
6	2(0.137)	3(0.065)	1(0.030)	7(0.023)	5(0.006)	*
6,2	3(0.088)	1(0.041)	—	—	—	*
6,2,3	1(0.080)	—	—	—	—	*
—	—	—	—	—	—	*

Table 7.13:

Melody 7: 1 3 5 5 4 2 3 1 1 3 5 5 4 2 3 1						
Working Memory	Expectations					Code
—	1(0.260)	3(0.125)	5(0.059)	5(0.028)	—	*
1	3(0.137)	5(0.065)	5(0.030)	1(0.010)	2(0.006)	*
1,3	5(0.087)	5(0.041)	2(0.014)	1(0.007)	4(0.006)	*
1,3,5	5(0.070)	—	—	—	—	*
—	4(0.252)	2(0.122)	3(0.057)	1(0.027)	—	*
4	2(0.133)	3(0.063)	1(0.029)	—	—	*
4,2	3(0.086)	1(0.040)	—	—	—	*
4,2,3	1(0.079)	—	—	—	—	*
—	1(0.260)	3(0.125)	5(0.059)	5(0.028)	—	*
1	3(0.137)	5(0.065)	5(0.030)	1(0.010)	2(0.006)	*
1,3	5(0.086)	5(0.041)	2(0.013)	1(0.006)	4(0.005)	*
1,3,5	5(0.069)	—	—	—	—	*
—	4(0.252)	2(0.122)	3(0.057)	1(0.027)	—	*
4	2(0.133)	3(0.063)	1(0.029)	—	—	*
4,2	3(0.086)	1(0.040)	—	—	—	*
4,2,3	1(0.079)	—	—	—	—	*
—	—	—	—	—	—	*

Table 7.14:

Melody 8: 5 5 3 1 2 2 $\underline{7}$ $\underline{5}$ 5 5 3 1 $\underline{6}$ $\underline{7}$ 1^2 3 4 3 1 2 2 $\underline{7}$ $\underline{5}$ 3 4 3 1 $\underline{6}$ $\underline{7}$ 1^2						
Working Memory	Expectations					Code
—	5(0.252)	5(0.125)	3(0.060)	1(0.028)	—	*
5	5(0.138)	3(0.066)	1(0.031)	4(0.023)	2(0.001)	*
5,5	3(0.090)	1(0.043)	—	—	—	*
5,5,3	1(0.083)	—	—	—	—	*
—	2(0.252)	2(0.122)	7(0.057)	5(0.027)	—	*
2	2(0.133)	7(0.063)	5(0.029)	1(0.011)	4(0.006)	*
2,2	7(0.085)	5(0.040)	—	—	—	*
2,2,7	5(0.078)	—	—	—	—	*
—	5(0.252)	5(0.125)	3(0.060)	1(0.028)	—	*
5	5(0.138)	3(0.066)	1(0.031)	4(0.023)	2(0.001)	*
5,5	3(0.090)	1(0.043)	—	—	—	*
5,5,3	1(0.083)	—	—	—	—	*
—	6(0.262)	7(0.125)	1(0.059)	—	—	*
6	7(0.172)	1(0.081)	5(0.006)	2(0.006)	3(0.003)	*
6,7	1(0.170)	—	—	—	—	*
—	3(0.263)	4(0.127)	3(0.060)	1(0.028)	—	*
3	4(0.139)	3(0.065)	1(0.031)	2(0.023)	5(0.010)	*
3,4	3(0.089)	5(0.047)	1(0.042)	2(0.014)	—	*
3,4,3	1(0.081)	—	—	—	—	*
—	2(0.252)	2(0.122)	7(0.057)	5(0.027)	—	*
2	2(0.133)	7(0.063)	5(0.029)	1(0.011)	4(0.006)	*
2,2	7(0.085)	5(0.040)	—	—	—	*
2,2,7	5(0.078)	—	—	—	—	*
—	3(0.263)	4(0.127)	3(0.060)	1(0.028)	—	*
3	4(0.139)	3(0.065)	1(0.031)	2(0.023)	5(0.010)	*
3,4	3(0.089)	5(0.048)	1(0.042)	2(0.014)	—	*
3,4,3	1(0.081)	—	—	—	—	*
—	6(0.263)	7(0.125)	1(0.059)	—	—	*
6	7(0.172)	1(0.081)	2(0.006)	5(0.006)	3(0.003)	*
6,7	1(0.170)	—	—	—	—	*
—	—	—	—	—	—	*

Table 7.15:

Melody 9: 1 5 6 5 4 3 2 1 $\underline{1}$ 7 6 5 $\underline{1}$ 5 3 1						
Working Memory	Expectations					Code
—	1(0.260)	5(0.131)	6(0.063)	5(0.030)	—	*
1	5(0.143)	6(0.069)	5(0.033)	3(0.023)	1(0.010)	*
1,5	6(0.093)	1(0.048)	5(0.044)	3(0.015)	—	*
1,5,6	5(0.084)	—	—	—	—	*
—	4(0.239)	3(0.119)	2(0.057)	1(0.027)	—	*
4	3(0.130)	2(0.063)	1(0.030)	—	—	*
4,3	2(0.086)	1(0.040)	—	—	—	*
4,3,2	1(0.078)	—	—	—	—	*
—	1(0.254)	7(0.122)	6(0.058)	5(0.027)	—	*
1	7(0.134)	6(0.063)	5(0.030)	3(0.022)	1(0.010)	*
1,7	6(0.084)	5(0.039)	1(0.012)	—	—	*
1,7,6	5(0.074)	—	—	—	—	*
—	1(0.263)	5(0.127)	3(0.061)	1(0.029)	—	*
1	5(0.140)	3(0.067)	1(0.032)	2(0.006)	7(0.006)	*
1,5	3(0.091)	1(0.047)	6(0.014)	5(0.007)	—	*
1,5,3	1(0.083)	—	—	—	—	*
—	—	—	—	—	—	*

Table 7.16:

Melody 10: 1 3 2 1 3 5 4 3 6 5 4 3 1 3 2 1						
Working Memory	Expectations					Code
—	1(0.250)	3(0.124)	2(0.060)	1(0.028)	—	*
1	3(0.136)	2(0.066)	1(0.031)	5(0.021)	7(0.005)	*
1,3	2(0.086)	1(0.040)	5(0.039)	5(0.006)	4(0.005)	*
1,3,2	1(0.077)	4(0.017)	—	—	—	*
—	3(0.262)	5(0.130)	4(0.062)	3(0.029)	—	*
3	5(0.142)	4(0.068)	3(0.032)	2(0.023)	1(0.011)	*
3,5	4(0.093)	3(0.044)	—	—	—	*
3,5,4	3(0.084)	—	—	—	—	*
—	6(0.259)	5(0.122)	4(0.057)	3(0.026)	—	*
6	5(0.134)	4(0.063)	3(0.029)	7(0.023)	1(0.011)	*
6,5	4(0.086)	3(0.040)	—	—	—	*
6,5,4	3(0.080)	—	—	—	—	*
—	1(0.255)	3(0.124)	2(0.060)	1(0.028)	—	*
1	3(0.136)	2(0.066)	1(0.031)	5(0.021)	7(0.006)	*
1,3	2(0.089)	5(0.044)	1(0.042)	5(0.006)	4(0.005)	*
1,3,2	1(0.079)	4(0.017)	—	—	—	*
—	—	—	—	—	—	*

Table 7.17:

Melody 11: 5 3 4 2 1 3 5^2 5 3 4 2 1 5 1^2						
Working Memory	Expectations					Code
—	5(0.262)	3(0.130)	4(0.062)	2(0.029)	—	*
5	3(0.142)	4(0.068)	2(0.032)	5(0.006)	1(0.001)	*
5,3	4(0.092)	2(0.044)	—	—	—	*
5,3,4	2(0.083)	—	—	—	—	*
—	1(0.266)	3(0.127)	5(0.060)	—	—	*
1	3(0.174)	5(0.082)	1(0.010)	2(0.006)	7(0.005)	*
1,3	5(0.145)	2(0.011)	5(0.005)	1(0.005)	4(0.005)	*
—	5(0.262)	3(0.130)	4(0.062)	2(0.029)	—	*
5	3(0.142)	4(0.068)	2(0.032)	5(0.006)	1(0.001)	*
5,3	4(0.092)	2(0.044)	—	—	—	*
5,3,4	2(0.083)	—	—	—	—	*
—	1(0.237)	5(0.118)	1(0.057)	—	—	*
1	5(0.162)	1(0.078)	3(0.023)	2(0.006)	7(0.005)	*
1,5	1(0.162)	3(0.013)	6(0.013)	5(0.006)	—	*
—	—	—	—	—	—	*

Table 7.18:

Melody 12: 1 2 3 1 1 2 3 1 3 4 5^2 3 4 5^2 5 5 3 1 5 5 3 1 1 $\underline{5}$ 1^2 1 $\underline{5}$ 1^2						
Working Memory	Expectations					Code
—	1(0.260)	2(0.125)	3(0.059)	1(0.028)	—	*
1	2(0.137)	3(0.064)	1(0.030)	5(0.021)	7(0.006)	*
1,2	3(0.088)	1(0.041)	—	—	—	*
1,2,3	1(0.080)	—	—	—	—	*
—	1(0.260)	2(0.125)	3(0.059)	1(0.028)	—	*
1	2(0.137)	3(0.064)	1(0.030)	5(0.021)	7(0.006)	*
1,2	3(0.087)	1(0.041)	—	—	—	*
1,2,3	1(0.079)	—	—	—	—	*
—	3(0.233)	4(0.116)	5(0.056)	—	—	*
3	4(0.159)	5(0.076)	2(0.022)	1(0.011)	3(0.003)	*
3,4	5(0.158)	2(0.012)	3(0.012)	1(0.006)	—	*
—	3(0.233)	4(0.116)	5(0.056)	6(0.000)	7(0.000)	*
3	4(0.159)	5(0.076)	2(0.022)	1(0.011)	3(0.003)	*
3,4	5(0.158)	2(0.012)	3(0.012)	1(0.006)	—	*
—	5(0.252)	5(0.125)	3(0.060)	1(0.028)	—	*
5	5(0.138)	3(0.066)	1(0.031)	4(0.023)	2(0.001)	*
5,5	3(0.090)	1(0.043)	—	—	—	*
5,5,3	1(0.083)	—	—	—	—	*
—	5(0.252)	5(0.125)	3(0.060)	1(0.028)	—	*
5	5(0.138)	3(0.066)	1(0.031)	4(0.023)	2(0.001)	*
5,5	3(0.090)	1(0.043)	—	—	—	*
5,5,3	1(0.083)	—	—	—	—	*
—	1(0.237)	5(0.118)	1(0.057)	—	—	*
1	5(0.162)	1(0.078)	3(0.023)	2(0.006)	7(0.005)	*
1,5	1(0.162)	3(0.013)	6(0.013)	5(0.006)	—	*
—	1(0.237)	5(0.118)	1(0.057)	—	—	*
1	5(0.162)	1(0.078)	3(0.023)	2(0.006)	7(0.005)	*
1,5	1(0.162)	3(0.013)	6(0.013)	5(0.006)	—	*
—	—	—	—	—	—	*

Table 7.19:

Input: 1 2 3 1 1 2 3 1					
Working Memory	Expectations				
1	3(0.023)	5(0.021)	1(0.010)	2(0.006)	7(0.006)
1,2	3(0.014)	1(0.007)	—	—	—
1,2,3	1(0.026)	—	—	—	—
1,2,3,1	—	—	—	—	—
1,2,3,1,1	—	—	—	—	—
1,2	3(0.087)	1(0.041)	—	—	—
1,2,3	1(0.080)	—	—	—	—
—	1(0.260)	3(0.233)	2(0.125)	4(0.116)	5(0.056)

Table 7.20:

which high-level d-cells are activated by the available evidence, evidence that can stem from the prior presentation of items or from the alternative sources described above.

Finally, the fact that ambiguity can exist at the level of phrase expectations and that this ambiguity can manifest itself at the note level, is illustrated in Table 7.20. This shows the expectations generated when the network is presented with the sequence $(1, 2, 3, 1, 1, 2, 3, 1)$ *without* prior identification of the melody. There is a delay in chunking out the first phrase, since it is not expected, but no delay in chunking out the second phrase, since it is expected. Having chunked out the second phrase, however, there is approximately equal expectation for either a "1" or a "3". This is because both Melody 6 and Melody 12 begin with the two phrases $(1, 2, 3, 1)$ and $(1, 2, 3, 1)$ but thereafter diverge, one continuing with a "1", the other continuing with a "3". As soon as one or other of these continuations occurs there is no further ambiguity in the expectations generated — the identity of the melody is unambiguous.

7.4 Summary

In this chapter I have detailed the modifications necessary to allow **SONNET 1** to handle repeated items and to cope with extended sequences. I have described the way in which these were implemented and how the resulting network was applied to the unsupervised learning of a set of simple melodies. Mechanisms for both recognition and recall of a familiar melody were discussed, and simulations confirmed their validity. A distinction was made between veridical and schematic expectations, concerning the activation, or lack of activation, of high-level d-cells.

Whilst the correct sequence of items can be recalled in the manner described above, these items will not necessarily be performed in the appropriate rhythm. In the next chapter, I will suggest, *inter alia*, how this can be achieved.

Chapter 8

Modelling aspects of rhythmic structure

8.1 Introduction

This chapter describes some work relating to the modelling of the rhythmic structure of music. The word "rhythmic" is here used in its most general sense, since the chapter is concerned with a range of connected issues, including the ability of piece of music to induce in a listener the awareness of a "beat", the relationship of this beat to a larger metrical hierarchy and the (limited) invariance to tempo found in the recognition and production of rhythmic patterns. It was decided to take as a starting point one of the places at which temporal/rhythmic aspects interact most obviously with the **SONNET** sequential memory described in earlier chapters. An attempt is made to place the perception of various components of rhythmic structure within a "neural" framework consistent with that adopted previously.

8.2 The need for a rhythmic pulse generator in SONNET

The motive for much of the work in the early part of this chapter comes from a design constraint that was placed on the operation of the **SONNET** sequential memory, namely that the operation of a **SONNET** module should not be adversely affected by variation in the absolute rate at which events occur at the lower level, $\mathcal{F}_1$. This constraint was seen to be

particularly important when considering the operation of a hierarchy of **SONNET** modules, there necessarily being a large variation between the "event rates" found at different levels of the hierarchy. Insensitivity to the absolute event rate allows a common set of (operational) parameters to be used for all **SONNET** modules, regardless of their position in the hierarchy. The suggested solution involved postulating that changes in activities and weights in a given module should only be enabled for a fixed time period after the onset of an event at the module's $\mathcal{F}_1$. This was shown to be an effective way of achieving the desired insensitivity to event rate. Unfortunately it had an undesirable side-effect, namely that it made the STM pattern at $\mathcal{F}_1$ insensitive to the relative *duration* (defined as IOI) of events. Thus the sequence $(1, 2, 3^2, 4, 5, 6)$ gives rise to the same STM pattern as the sequence $(1, 2, 3, 4, 5, 6)$. Moreover, the suppression of any "additional" processing in between the onsets of events "3" and "4" in the first sequence hinders the early "chunking out" of the first phrase (assuming that it is familiar enough to be represented at $\mathcal{F}_2$). It also prevents a related effect in the operation of LTM, identified as useful in previous chapters, whereby phrases consistently ending in relatively long notes (most often tonic triad "resting" notes) enjoy enhanced LTM learning.

To avoid these undesirable side-effects it was suggested that changes at a given module should not only be enabled for a short time after the onset of an event at $\mathcal{F}_1$ but also for a short time after an *anticipated* event onset at this level. This anticipation results from the assumption that a stimulus consisting of a series of events regularly spaced in time will be likely to continue to exhibit this regularity in the immediate future. To illustrate the point, we imagine that a series of regularly spaced events induces synchronized activity in a listener, possibly manifesting itself physically as, for instance, the tapping of a foot. Activity changes in a **SONNET** module could then be seen as being enabled for a fixed time period after the occurrence of each foot-tap, rather than just after the occurrence of an actual event. The foot-tapping can thus be seen as the physical manifestation of a rhythmic attentional pulse which is able to synchronize itself with an environmental regularity and, once synchronized, persist for *short* periods in the absence of actual environmental confirmation. Clearly a *continued* absence of confirmatory events should reduce the ability of the attentional pulse to sustain itself.

A rhythmic attentional pulse can be used in another way, namely to highlight important structural notes in a given melody. There is considerable evidence that notes sounded

at rhythmically prominent time-points are better stored in short-term memory than notes sounded at less prominent times (see, for instance, the work of Jones discussed later in this chapter). Since, in the framework developed previously, we assume that the STM pattern is instrumental in eliciting information from long-term memory, in the form of expectations etc., we can see that the rhythmic structure of a sequential stimulus, as well as its pitch content, will have an effect on the precise long-term memories accessed. This was implicit in the "simplification" applied to the melodies used in earlier chapters, which involved identifying and using only those notes which fell "on the beat", thus isolating the sequential memory from confusions which might result from melodic embellishments, such as trills or quick note-repetitions. The simplification process can therefore be thought of as the filtering of the stimulus by the rhythmic attentional pulse. As noted earlier, such an approach reveals the equivalence of several well-known nursery rhymes, e.g. "The Alphabet Song" and "Twinkle, Twinkle, Little Star", which differ only in their melodic embellishments (these differences being necessary to accommodate the verbal accompaniment). Any sequential memory which did not apply some level of attentional filtering would be forced to store the melodic structure of each of these two tunes separately, a strategy which would seem to be far from optimal.

Gjerdingen [62] adopts an approach similar to that described above by allowing an indicator of metrical strength to modulate the attention paid to events at the input to his "L'ART pour l'art" system (discussed in my Chapter 3). He chooses to grade the attention paid to notes at differing metrical positions in a manner more appropriate to the requirements of his network (which is not a sequential memory), but his scheme is essentially the same as that used here. In the same paper, Gjerdingen postulates the existence of "neural metronomes", which can be used to generate the rhythmic attentional pulse described above. These consist of small neural circuits whose topology and parameters are such that they exhibit a rhythmically changing pattern of activity. The model described in the next section is similar to Gjerdingen's, but is based on a neural circuit, the "gated pacemaker", developed by Carpenter and Grossberg [21] as part of a theory relating to circadian rhythms. The use of this circuit has some advantages over Gjerdingen's approach; in particular, the way in which the pacemaker "tracks" an environmental stimulus can be made explicit.

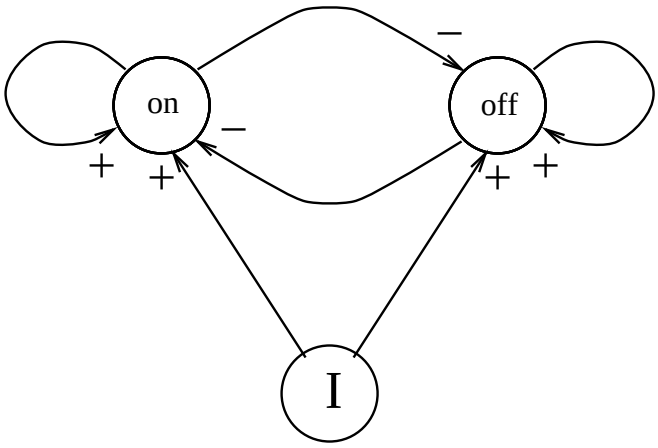


Figure 8.1: The gated pacemaker

8.3 The gated pacemaker model

The gated pacemaker is a very simple neural circuit consisting, in its most basic form, of only three cells. Two of the cells, the so-called "on-cell" and "off-cell", are arranged as a gated-dipole, that is each excites itself and inhibits the other. In addition both cells receive a tonic arousal signal. The arrangement is shown in Figure 8.1. The self-excitatory signals are each gated by a transmitter substance, the quantity of which varies under the influence of antagonistic processes which deplete it and re-accumulate it at rates dependent on a number of factors. The following equations fully describe the variation of cell activities and transmitter quantities

$$\frac{d}{dt}X_{on} = -AX_{on} + (B - X_{on})[I + F(X_{on})Z_{on}] - (X_{on} + C)G(X_{off}) \tag{8.1}$$

$$\frac{d}{dt}X_{off} = -AX_{off} + (B - X_{off})[I + F(X_{off})Z_{off}] - (X_{off} + C)G(X_{on}) \tag{8.2}$$

$$\frac{d}{dt}Z_{on} = D(E - Z_{on}) - KF(X_{on})Z_{on} \tag{8.3}$$

$$\frac{d}{dt}Z_{off} = D(E - Z_{off}) - KF(X_{off})Z_{off} \tag{8.4}$$

where A, B, C, D, E and K are constants, X_{on} and X_{off} are the activities of the on-cell and off-cell respectively and Z_{on} and Z_{off} represent the amount of gating transmitter at each. The

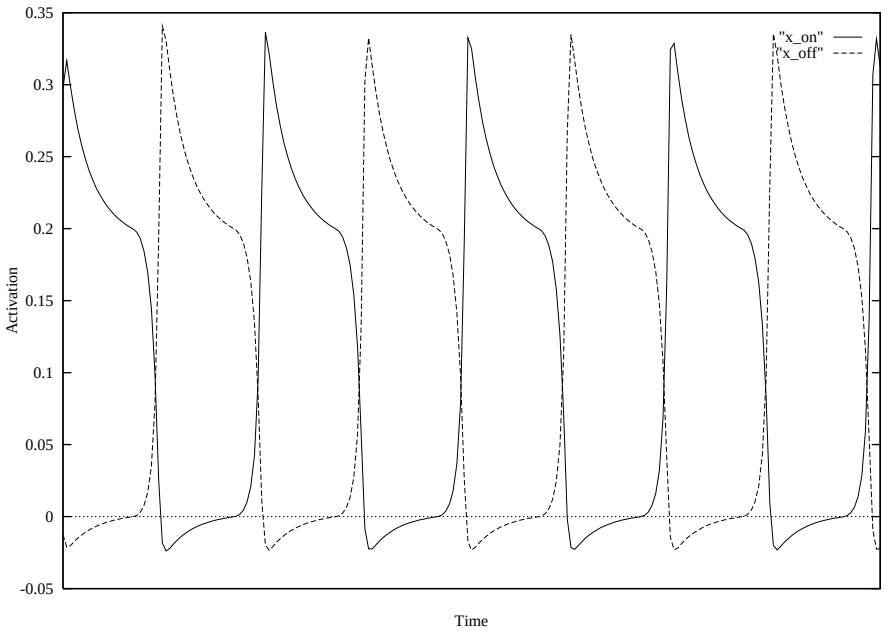


Figure 8.2: "Flip-Flop" oscillation of gated pacemaker activities.

functions F and G can be sigmoid or threshold linear (they are assumed to be the latter in the work presented here). The differential equations describing the change in cell activities with time are the familiar equations for a cell with "shunting" dynamics, used throughout this thesis, whilst the transmitter equations are essentially the same as those used in the **ART 3** network described in Chapter 4. The transmitter is accumulated towards its maximum level, E, at a rate D and is simultaneously depleted at a rate $KF(X)Z$, as it is used in the self-excitation process. To clarify the operation of the model Carpenter and Grossberg describe dimensionless versions of these equations, which have the form

$$\frac{d}{dt}x_{on} = -x_{on} + (1 - x_{on})[C_1 + C_2 f(x_{on})z_{on}] - (x_{on} + C_3)C_4 g(x_{off}) \tag{8.5}$$

$$\frac{d}{dt}z = C_5[1 - z_{on} - C_6 f(x_{on})z_{on}] \tag{8.6}$$

This arrangement of cells results (for a range of parameter choices) in the "flip-flop" oscillation of cell activities as shown in Figure 8.2. Briefly, we suppose that both cells begin with their transmitter fully accumulated but with different levels of activity, e.g. x_{on} large, x_{off} small. For a time x_{on} is able to stay high, thus keeping x_{off} low by its inhibitory action. While x_{on} is high, however, its self-excitatory transmitter is being depleted faster than it can be re-

accumulated and, as a result, the self-excitatory effect weakens. By contrast the transmitter z_{off} remains fully accumulated (by virtue of the low value of x_{off}) so that the potential for self-excitation at this cell remains high. Eventually, the weakening in the self-excitation at the on-cell causes its activity to drop, which in turn reduces the lateral inhibition to the off-cell, causing x_{off} to rise. The off-cell now has an advantage in the competition for activation that ensues and the relative cell activations quickly reverse. The particular manifestation of the gated pacemaker used here is entirely "symmetrical" (though Carpenter and Grossberg do suggest unsymmetrical versions for specific applications) and so the same cycle begins again, this time with x_{off} initially high, x_{on} low. In a series of simulations Carpenter and Grossberg show that the period of the oscillation is primarily governed by the parameter C_5.

Although the gated pacemaker model was developed to explain data relating to circadian rhythms, there is nothing that restricts its application to periods of a particular absolute value. Indeed Carpenter and Grossberg summarize

> "...the same gated pacemaker circuit can be made to generate rhythms of essentially any period...thus the parametric studies reported herein may find their way into discussions of fast perceptual and cognitive rhythmic phenomena that seem to be unrelated to circadian rhythms, but on a mechanistic level may be properties of gated pacemaker circuitry."

Grossberg [69] postulates that there are at least four distinct mechanisms underlying the human "rhythmic capability", one of which involves the gated pacemaker circuit described above. The model he suggests seems ideally suited to function as the generator of a rhythmic attentional pulse. He postulates that a rhythmic stimulus, acting as input to *one* of the competing cells, will initiate the gated pacemaker oscillation and that feedback from the competing cells to the tonic arousal cell will then quickly determine the level at which the arousal cell should be maintained if it is to sustain the oscillations at the same rate in the absence of the stimulus. Furthermore, he adds that the arousal cell activation can subsequently be altered by an "act of will" so as to speed up, slow down or suppress the oscillations. Unfortunately, there are several reasons why such an explanation appears to be inadequate. Firstly Carpenter and Grossberg [21], in their investigation of various gated pacemaker parameter choices, show that the period of the oscillator is "relatively insensitive to the dimensionless arousal level". This suggests the implausibility of any arrangement in which feedback to the

arousal source is responsible for "tuning" the period of the oscillator to that of the stimulus. Secondly, the mechanism suggested above requires that the faster the stimulus, the higher should be the feedback to the tonic arousal cell. Once again the simulations of Carpenter and Grossberg [21] contradict this interpretation by showing that a shorter oscillation period actually leads to a lower average level of activity at the competing cells. In spite of these contradictions, the basic idea, i.e. that a gated pacemaker can quickly synchronize itself with a rhythmic stimulus, is an appealing one. In the next section modifications are made to the basic pacemaker circuit to enable this result to be achieved.

8.4 Synchronizing a pacemaker with a rhythmic stimulus

8.4.1 Formalizing the problem

Before showing how the pacemaker can be synchronized with a stimulus, it is necessary to formalize the problem by making specific our requirements for the resulting circuit. Firstly we assume that the stimulus signal (i.e the signal that stimulates the pacemaker) is defined as the output, or equivalently the activation (assuming a linear output response), of a latch cell, similar to that used in the networks of previous chapters, which turns on in response to the onset of an environmental event and remains on for a fixed amount of time before switching off. Thus a series of regularly spaced stimulus onsets can be represented by a rectangular wave (latch cell activation versus time) with a mark-space ratio dependent on the stimulus period. Likewise, we assume that a similar signal can be derived from the gated pacemaker, i.e. that an equivalent latch cell is connected to each (or conceivably one) of the on-cell and off-cell—in each case the latch cell turns on at a given point on the single-cycle activity-versus-time plot of its corresponding cell. For definiteness we assume that the latch cell fires at a given point on the rising slope of the activity plot as shown in Figure 8.3. In this way the gated pacemaker also generates a square wave defined by the sum of the outputs of the attached latch cells (see Figure 8.4). For simplicity we assume that the stimulus and pacemaker latch cells have the same characteristics, though this is not necessary in the following analysis. Synchronization between stimulus and pacemaker is then simple defined as the synchronization of the two

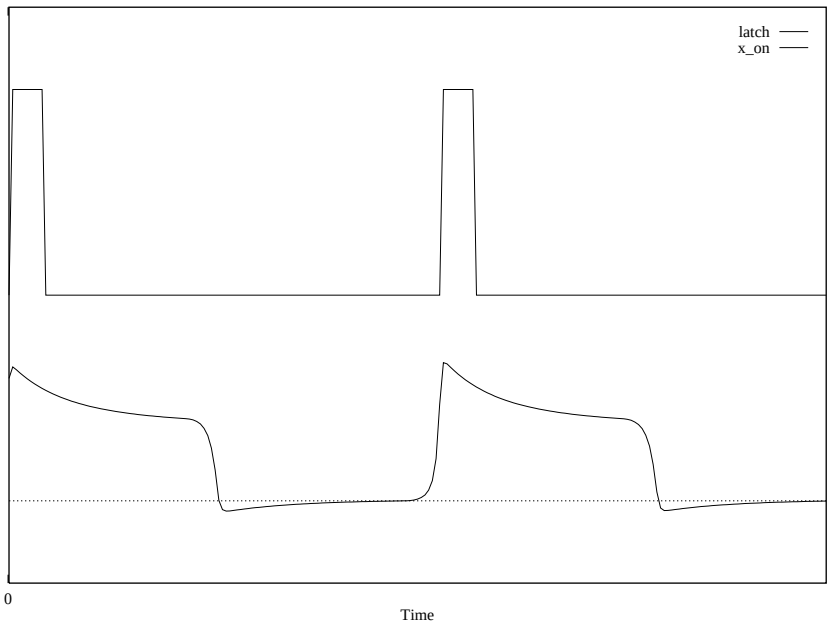


Figure 8.3: Latch responds to rising edge of activity plot.

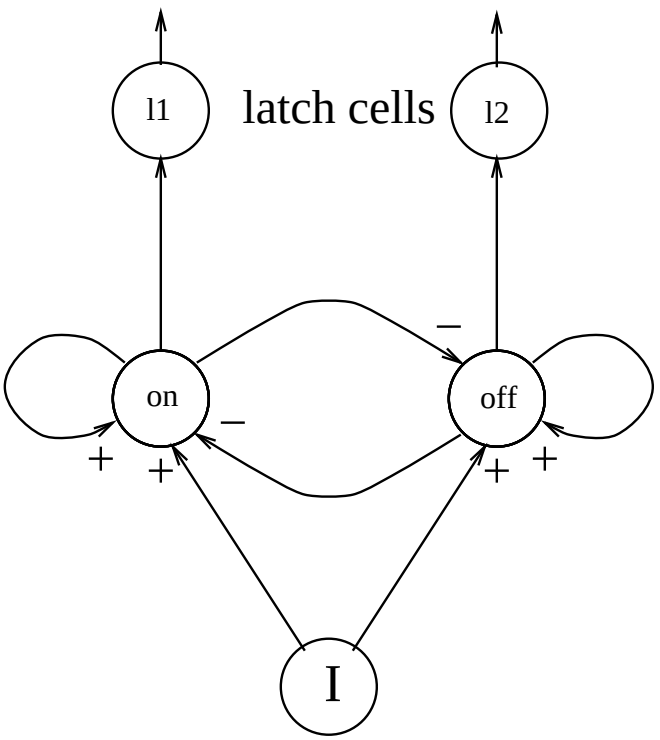


Figure 8.4: A latch cell is connected to each pacemaker cell.

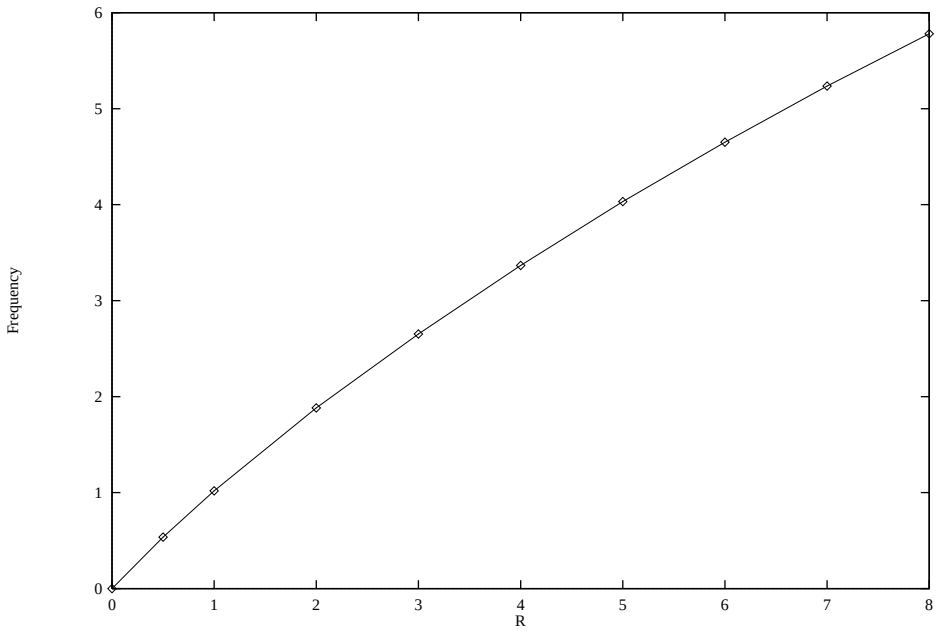


Figure 8.5: The variation of pacemaker frquency with R.

square waves. It should be noted that this requires not only that the pacemaker response should acquire the same *period* as the stimulus but also that it should be *in phase* with the stimulus.

The second assumption we shall make is that there exists an additional cell, whose output (or activity), R, replaces the parameter C_5 in the differential equation describing changes in the level of gating transmitter, z. Equation 8.6 now becomes

$$\frac{d}{dt}z = R[1 - z - C_6 f(x)z] \tag{8.7}$$

Thus the activity R modulates the rate of change in the amount of available gating transmitter and, as such, is capable of controlling the period of the oscillator. The cell with activity R will therefore be referred to as the pacemaker's rate-determining cell. Figure 8.5 shows that an approximately linear relationship holds between the value of R and the oscillator frequency—this property proves useful later in the chapter. The rate-determining cell now plays the rôle of the arousal cell in Grossberg's suggested scheme, whilst the arousal cell continues to deliver a constant tonic input to both on-cell and off-cell. The modified circuit is shown in Figure 8.6.

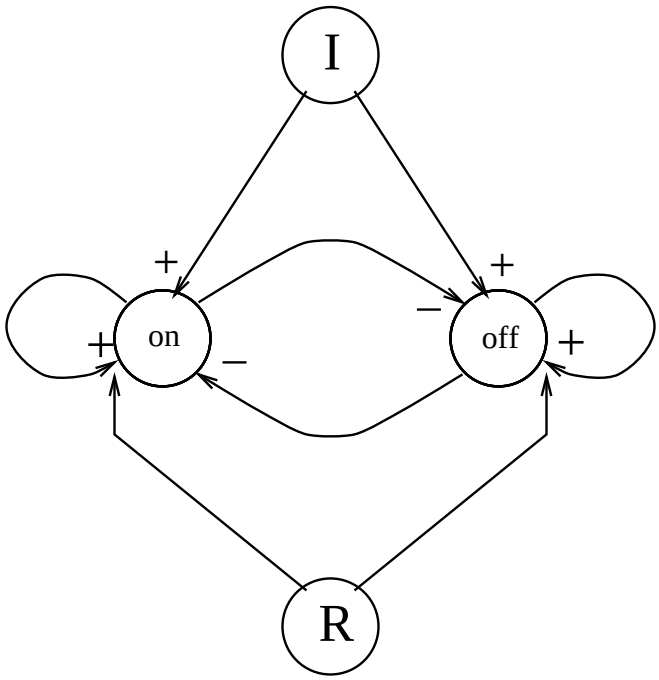


Figure 8.6: The activity R modulates changes in gating transmitter and controls the oscillation period (latch cells not shown).

8.4.2 The phase-locked loop

The further development of the circuit is motivated by the observation that its desired operation is equivalent to that of an electronic circuit commonly used in communications applications, the phase-locked loop (PLL). The block diagram of a PLL is shown in Figure 8.7. The phase detector generates a signal proportional to the phase error between the input signal and the signal produced by the voltage-controlled oscillator (VCO). This error signal is low-pass filtered and fed to the input of the VCO. The feedback loop is arranged so that the error signal tends to act so as to bring the output of the VCO into frequency and phase lock with the input signal. The design of a PLL primarily involves the choice of an appropriate phase detector and the placing of constraints on the feedback parameters to ensure stability. There are two common types of phase detector: a type I detector is simply an exclusive-or gate with the stimulus signal as one input and the response signal (i.e. the latched pacemaker output) as the other; the type II detector is sensitive to the relative timing of the upward-going edges of the input signals. The type II detector is more appropriate for use in the current application since it is insensitive to the mark-space ratio of the input signals (which can vary with signal period as noted above). The output characteristics of a type II detector are shown in

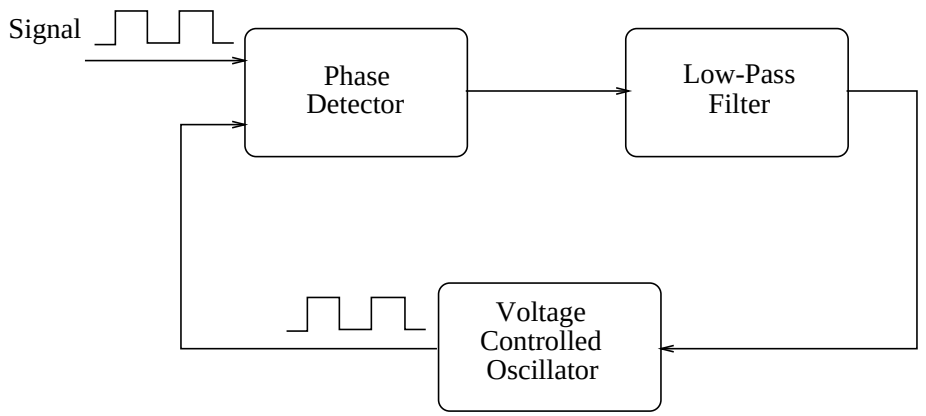


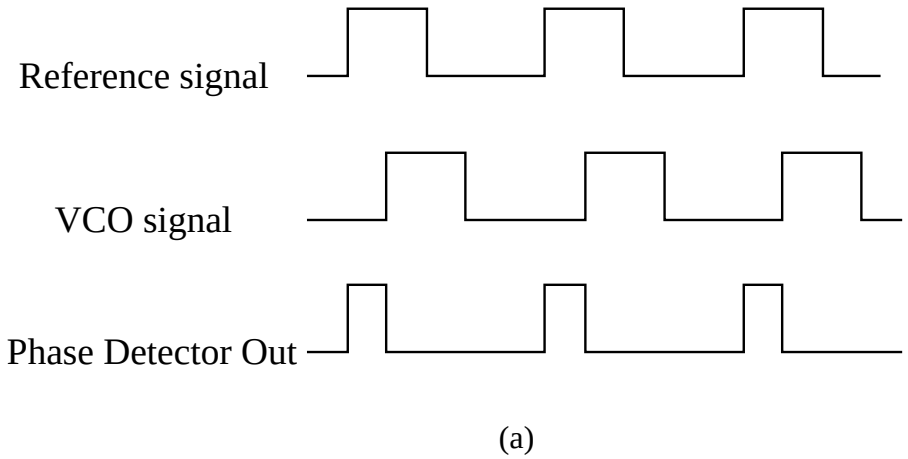
Figure 8.7: Block diagram of a phase-locked loop (PLL).

Figure 8.8. It can be seen that the sign of the phase detector output (also referred to as the error signal) depends on whether the stimulus lags or leads the response. The loop stability constraints are discussed in detail in Horowitz and Hill [76], who show that the stability of the PLL depends on the feedback "phase margin". The simple low-pass filter circuit shown in Figure 8.9 can be used (although Horowitz and Hill recommend a more complex "lead-lag" filter for more general applications). The values for resistance and capacitance are calculated by deciding upon a suitable value for the frequency at unity loop gain—in general a higher unity-gain frequency leads to faster tracking of the stimulus signal by the response signal.

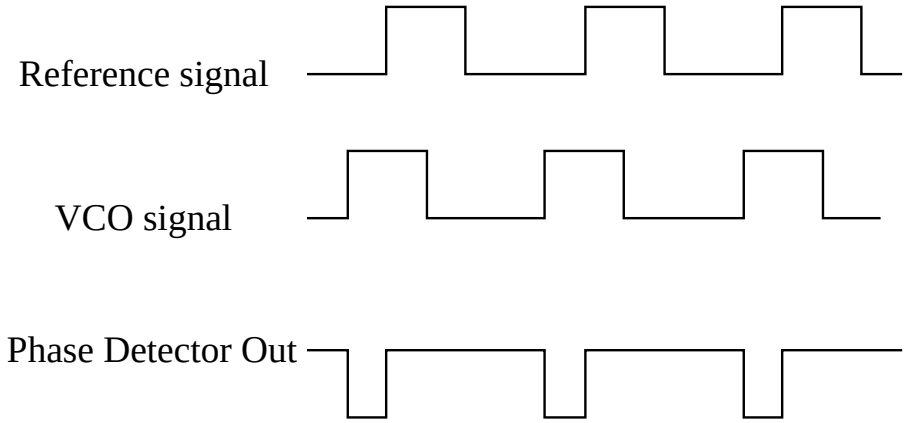
The similarity between the operation of the PLL and the functionality required of the attentional pulse generator is obvious. The stimulus and response signals have been re-cast as square wave signals and the gated pacemaker, with its period controlled by the activity R, is essentially equivalent to a voltage controlled oscillator. The next task was to provide "neural equivalents" of the two remaining components, the phase detector and the low-pass filter.

8.5 A neural phase-locked loop

Analysis of the low-pass filter circuit shown in Figure 8.9, shows that the following relationship holds between the input voltage, V_{in}, and the output voltage, V_{out}



(a)



(b)

Figure 8.8: The output characteristics of a type II phase detector for (a) VCO lags reference and (b) VCO leads reference.

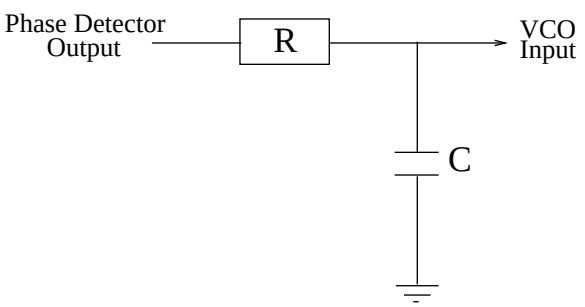


Figure 8.9: A simple low-pass filter.

$$\frac{d}{dt}V_{out} = \frac{1}{RC}\left[V_{in} - V_{out}\right] \tag{8.8}$$

The key to the development of a neural equivalent is the realization that the differential equation above is of the same form as the differential equation for the activation of a standard neuron with additive dynamics ([71][97])

$$\frac{d}{dt}x = -Ax + P - Q \tag{8.9}$$

where A is the decay parameter, P is the positive input and Q is the negative input to the cell. The equivalence is seen if we substitute V_{out} for x, $\frac{1}{RC}$ for A and $\frac{V_{in}}{RC}$ for P, setting Q to zero. Since in this case the low-pass filter is considered to be "open-circuit" when its input is zero, it is necessary to modulate the decay parameter, A, so that $A = 0$ at times when no input is present. Thus a standard neuron with additive dynamics can be configured as a low-pass filter by the choosing appropriate parameters—the activation of the neuron represents a low-pass-filtered version of its input signal. Since the filter output, V_{out}, controls the VCO, we can further substitute activity, R, for V_{out}, so that Equation 8.8 describes the change in activation of the pacemaker's rate-determining cell over time.

Figure 8.10 shows a simple arrangement of cells which acts like a type II phase detector—it is shown generating an input signal to the pacemaker's rate-determining cell, R, as required. The circuit consists of a matched pair of cells, labelled X_1 and X_2, whose activities can vary between zero and an upper limit (saturation value) of, say, one. These cells are influenced by the four latch cells labelled L_1, L_2, L_3 and L_4. Latch cells L_1 and L_4 turn on for a short time in response to rising edges in the stimulus and response signals respectively, whilst latch cells L_3 and L_2 do likewise providing they also receive input from the cell Y. Cell Y, in turn, is only active if both X_1 and X_2 have activations above a positive threshold of, say, 0.5. The activations X_1 and X_2 switch rapidly between their upper and lower limits when driven by the inputs from the latch cells, e.g. if $X_1 = 0$ and a stimulus onset is detected then the resulting positive input quickly drives X_1 to its saturation level. The net input, I_R, to the rate-determining cell, R, is simply given by $I_R = X_1 - X_2$. I_R can be substituted for V_{in} in Equation 8.8 which, together with the substitution of R for V_{out}, becomes

$$\frac{d}{dt}R = \frac{1}{D}\left[I_R - R\right] \tag{8.10}$$

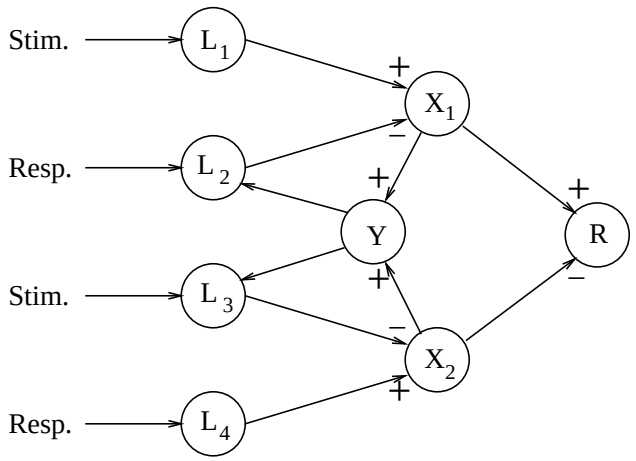


Figure 8.10: A "neural" type II phase detector.

where D represents a "damping" parameter equal to the LPF time constant.

The full circuit for the neural phase-locked loop is shown in Figure 8.11. Whilst no claim is made that this precise circuit exists *in vivo*, it is at least an example of a simple neural circuit, made up of a small number of "standard" neurons, which is functionally equivalent to the electronic PLL.

In the next section the performance of this network will be assessed.

8.6 Initial results

Simulations of the "neural" phase-locked loop were carried out. The operating region for the neural oscillator is assumed to be that shown in Figure 8.5. For values of R in the range $R = 0$ to $R = 8$, the frequency (f) varies in a near-linear fashion between $f = 0$ and $f \approx 6$ *units*. This operating range was chosen since, if the units are thought of as Hertz, it encompasses the range of response (e.g. tapping) frequencies typically elicited by rhythmic musical stimuli ([59],etc., see my Section 8.7). It should be noted that, although the pacemaker model was introduced in the context of circadian rhythms, the time axis can be "re-scaled" by the provision of a single multiplicative (capacitance) term in the relevant differential equations (Equations 8.5 and 8.6). In all the simulations it is assumed that the pacemaker output

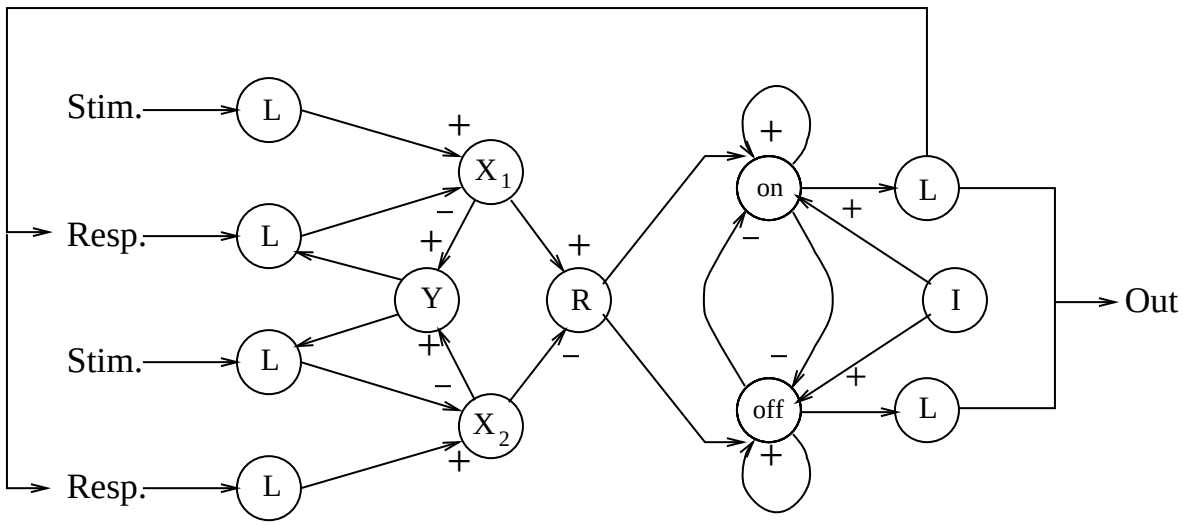


Figure 8.11: The full circuit for the neural PLL.

starts "in phase" with the stimulus signal, in the sense that the first pacemaker output pulse occurs simultaneously with a stimulus pulse. This can be thought of as being due to the action of a brief but large signal, which selectively excites one type of pacemaker cell (either the "on-cell" or the "off-cell") and which coincides with the first stimulus pulse that occurs after an "instruction" to track the stimulus. Although the stimulus and response signals start simultaneously, they will thereafter begin to generate phase detector error signals (except in the unlikely event that the initial value of R is exactly that needed to drive the pacemaker at the stimulus frequency). It is the job of these error signals to affect the value of R so as to bring the stimulus and response signals into phase lock. To enable the phase detector output to "pull" the value of R to any point in the desired operating range $0 < R < 8$, the phase detector cells, X_1 and X_2, are now assumed to saturate at activity $X = 10$ and the activation threshold for cell Y is adjusted accordingly. The value of R is similarly constrained to lie between $R = 0$ and $R = 10$.

Initial simulations highlighted some important problems with the operation of the circuit. The most important of these is illustrated in Figure 8.12, which shows the behaviour of the stimulus and response signals, together with the error signal and the changing value of R, as the loop attempts to bring the value of R down from an initial value of $R = 5$ to the value indicated by the stimulus period, in this case $R = 3$.

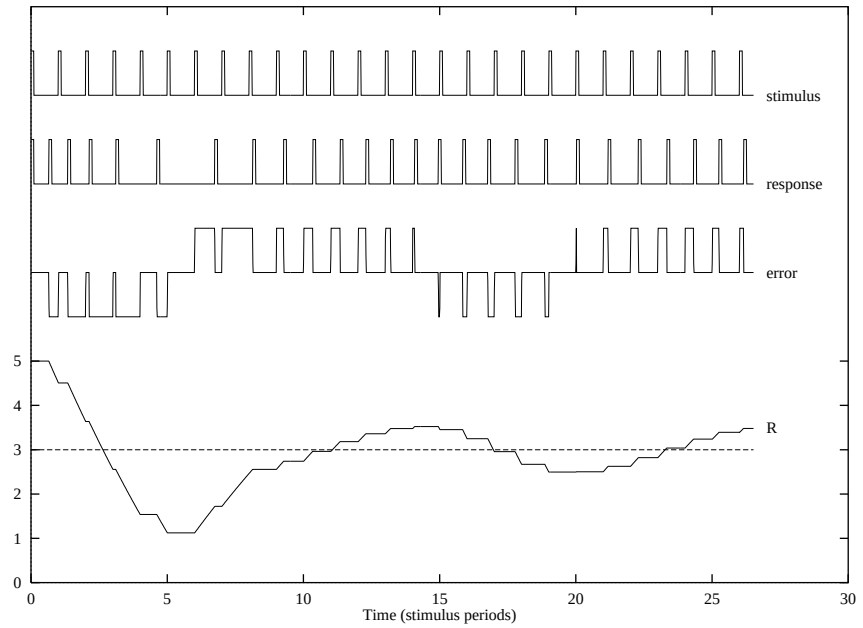


Figure 8.12: The pacemaker response for D=5.

As noted above, the first response pulse is coincident with the first stimulus pulse, whilst the second response pulse precedes the stimulus, the value of R being "too high". This generates an error signal which correctly causes the value of R to drop at a rate set by the choice of parameters in Equation 8.10. In the example shown, the rate of change of R under the influence of an error signal is set quite high (damping value, $D = 5$) in the hope of rapid convergence to the appropriate value. The fall in the value of R continues until the response signal has slowed sufficiently that *corresponding* pulses in the two signals are once more coincident. Unfortunately, by the time this occurs the value of R has *necessarily* overshot and is now lower than its target value. This results in a series of error signals which bring C_5 back up towards its target value. Once again, however, there is some overshoot and an oscillation around the target value ensues. This oscillation persists for a long time as is shown in Figure 8.13 — for the parameters used above, large oscillations continue up to and beyond a time equal to one hundred stimulus periods.

This is clearly an unacceptable level of performance. The problem of overshoot and oscillation can be affected by adjusting the damping parameter, D so as to reduce the rate of change of R under the influence of the error signal. Figures 8.14 shows the performance for various choices for the damping factor, D.

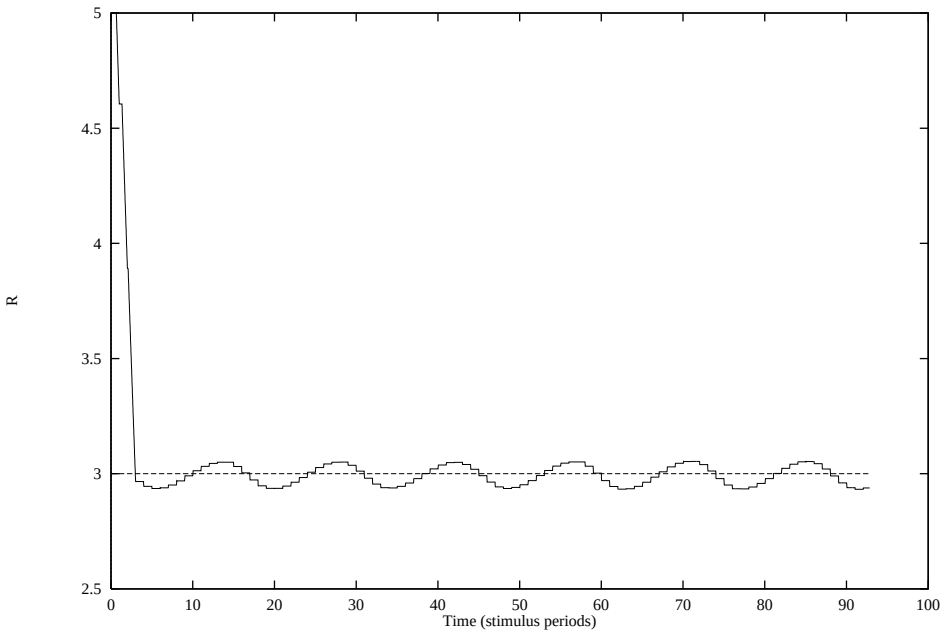


Figure 8.13: Persistent oscillation for D=5.

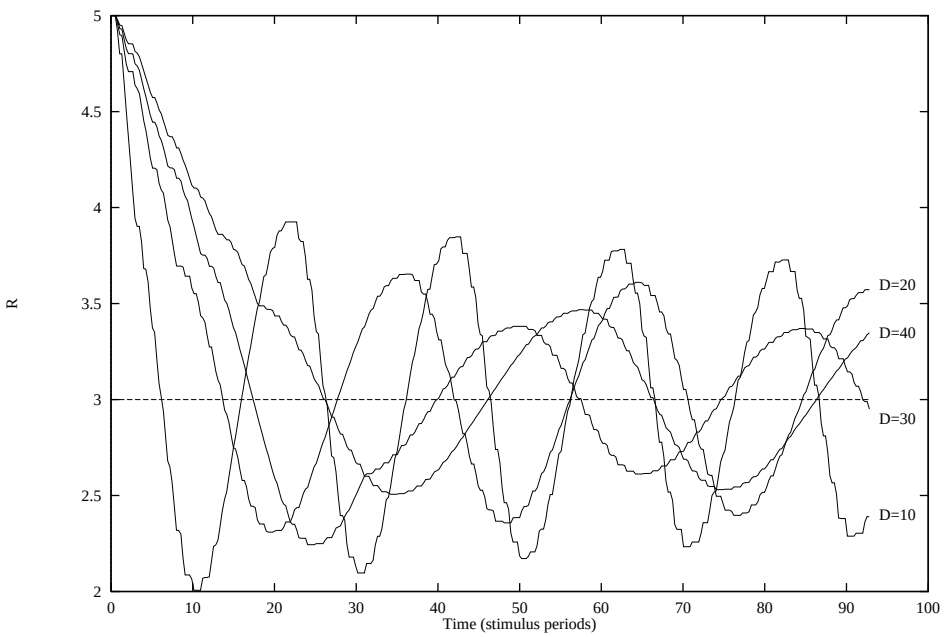


Figure 8.14: Persistent oscillations for various D.

As can be seen, damping of the oscillation cannot be achieved, even at the expense of a slowing of R's initial convergence to its target value— large oscillations in the value of R persist for the high damping ratios shown. It should also be noted that the magnitude of these oscillations does not decrease monotonically with an increase in D—clearly oscillations cannot be reliably prevented by judicious choice of the damping parameter.

To address this problem it was decided to use the error signal to "freeze" the state of the pacemaker. In particular, the pacemaker is frozen at all times when the error signal is negative —this is achieved by blocking (setting to zero) the gating action of R in Equation 8.7 at these times—note that the activity R is not itself set to zero but the effect it has on the gating transmitter is blocked, as if R were zero. This has the effect that, in the case of the response pulse *preceding* its corresponding stimulus pulse, the response signal is "held back" until the stimulus pulse arrives. During this "frozen" period, the value of R falls towards its target value as before. The advantage of freezing the pacemaker in this way is that, in the case where the error signal has brought the value of R close to its target value by the time the error signal turns off, the subsequent phase error will be correspondingly small. This is in contrast with the previous arrangement, in which the phase error could be large even when R had been brought close to its target value.

Unfortunately it is not practical to use a similar strategy in the case where the stimulus pulse *precedes* its corresponding response pulse. The stimulus pulse is a datum, so the only way of ensuring synchronization of the two pulse streams at the time of the stimulus onset would be to bring the response pulse forward in time. By doing so, however, the phase error signal, and the information that its duration provides regarding the adequacy of the current value of R, would be lost. It was therefore decided that this positive error signal should be permitted to increase R to a level above its target value (the overshoot described above), at which point the response pulses would once more be leading the stimulus pulses, allowing the "freeze" strategy to bring the signals into phase-lock without significant further overshoot. The success of this strategy in controlling oscillations, whilst achieving phase-lock in a reasonable time period, is illustrated in Figure 8.15. This shows the variation of R using the same initial/target conditions as before and with $D = 2$. It can be seen that the time taken to establish phase-lock is dramatically reduced when the "freeze" strategy is employed.

The second major problem with the circuit developed above is the insensitivity of the

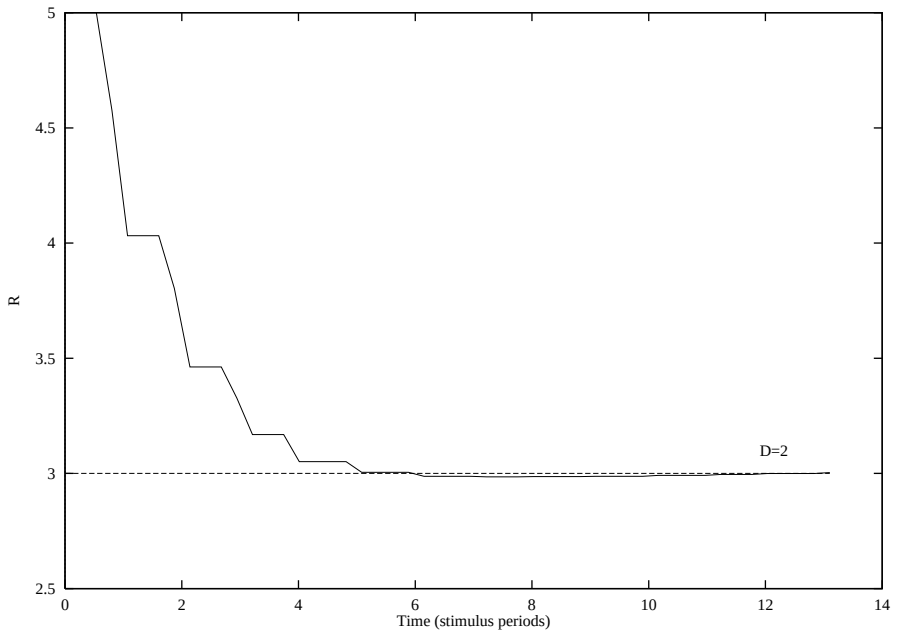


Figure 8.15: Variation in R using "freeze" strategy.

gradient term, describing the change in R with time under the influence of the error signal, to the value of R itself. Equation 8.10 shows that when the error signal is positive (i.e. the response lags the stimulus) R grows towards an asymptote equal to the magnitude of the error signal. When the error signal is negative the value of R falls towards a negative asymptote of the same magnitude (remember that R is limited to be greater than or equal to zero). Thus the magnitude of $\frac{d}{dt}R$ is always greater when R is falling. The problem occurs because the magnitude of this negative gradient is relatively insensitive to the current value of R—from Equation 8.10 we can see that it is equal to $\frac{1}{D}(R + |I_R|)$, where $|I_R|$ is the magnitude of the phase-detector output during error, in this case $|I_R| = 10$ implying that, for initial values $R = 5$ and $R = 2$, the initial gradients during error will be given by $\frac{d}{dt}R = \frac{15}{D}$ and $\frac{d}{dt}R = \frac{12}{D}$ respectively. Thus there is only a 20% drop in the magnitude of the initial negative gradient when the value of R is halved. This insensitivity of the gradient to the value of R can be seen to be a problem when one considers the amount of time for which the error signal acts in each case. For definiteness, consider the first time period after tracking begins and let the initial value of R be denoted R_i and the target value, indicated by the stimulus period, be denoted R_t ($R_t < R_i$). If we further assume that the frequency, f, of the pacemaker is related linearly to R so that $f = G.R$, where G is a constant (this is a reasonable approximation as can be

seen from Figure 8.5), then we can see that the error signal will be on for a time, t_e, where

$$t_e = \frac{1}{G}\left(\frac{1}{R_t} - \frac{1}{R_i}\right) \tag{8.11}$$

or equivalently

$$t_e = \frac{(R_i - R_t)}{G.R_i.R_t} \tag{8.12}$$

We now look again at the two cases considered above, the first in which $R_i = 5$ and $R_t = 4$, the second in which $R_i = 2$ and $R_t = 1$. From the equation above it can be seen that the times for which the error signal acts in each case are $t_e = \frac{1}{20G}$ and $t_e = \frac{1}{2G}$ respectively, i.e. the error signal acts for ten times as long for the latter case. Given the relatively small change in the magnitude of the error-induced gradient $\frac{d}{dt}R$ over this range of R values noted above, we can see that is impossible to choose a value for the damping parameter, D, which will ensure quick convergence of R to its target value in *both* the suggested cases: if the gradient is large enough to ensure near-convergence in the shorter time period then the application of the error signal over the longer time will cause severe overshoot; conversely, if the gradient is set to bring about near-convergence in the longer time period then it will be ineffective in doing so over the shorter period.

The solution to this problem lies in allowing the value of R to affect the error-induced gradient $\frac{d}{dt}R$. More particularly it can be shown that, in the case where the initial value of R is greater than its target value, convergence to the target value can be achieved in a single stimulus period for any such R_i and R_t by using the differential equation

$$\frac{d}{dt}R = e.G.R^2 \tag{8.13}$$

where e is the error term (phase-detector output), now once more constrained such that $-1 \leq e \leq 1$. This result assumes a perfectly linear relationship between R and the pacemaker frequency but, even given the relationship shown in Figure 8.5, the results for $R_i = 5$ and a variety of target values ($R_t < R_i$) are good and are shown in Figure 8.16. In each case convergence to within 2% of the target value was achieved within usually two and always three stimulus periods. Convergence in the case where $R_t > R_i$ is not quite so fast since the rise in the value of R once the error signal turns on brings the response pulse forward and reduces the time for which the error signal is able to act. Nevertheless, once the value of R

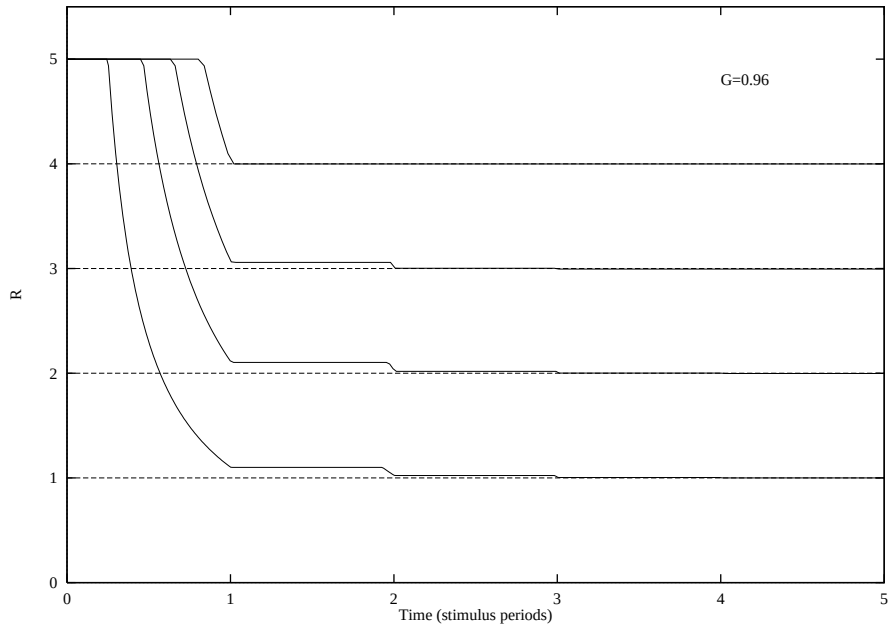


Figure 8.16: Variation in R for initial $R = 5$ and various target values, using freeze strategy and "recurrent" gradient term.

has been taken above its target value (sometimes quite a long way above), a combination of the "freeze strategy" and the error signal will bring about convergence within at most three further stimulus periods. The behaviour of the circuit is shown in Figure 8.17 for various cases in which $R_t > R_i$.

The fact that the change in the value of R is affected by the value of R itself implies "recurrent" connections in the corresponding neural circuit. Once again the activity of the pacemaker's rate-determining cell can be described by the standard equation for a neuron with additive dynamics

$$\frac{d}{dt}x = -Ax + P \tag{8.14}$$

in which the decay term, A, is zero, and the term P, representing net input, is given by the recurrent term

$$P = e.G.x^2 \tag{8.15}$$

where G is a constant and e represents the phase-detector output (phase error term), which modulates the sign of the recurrent input as it swings between plus and minus one. It should

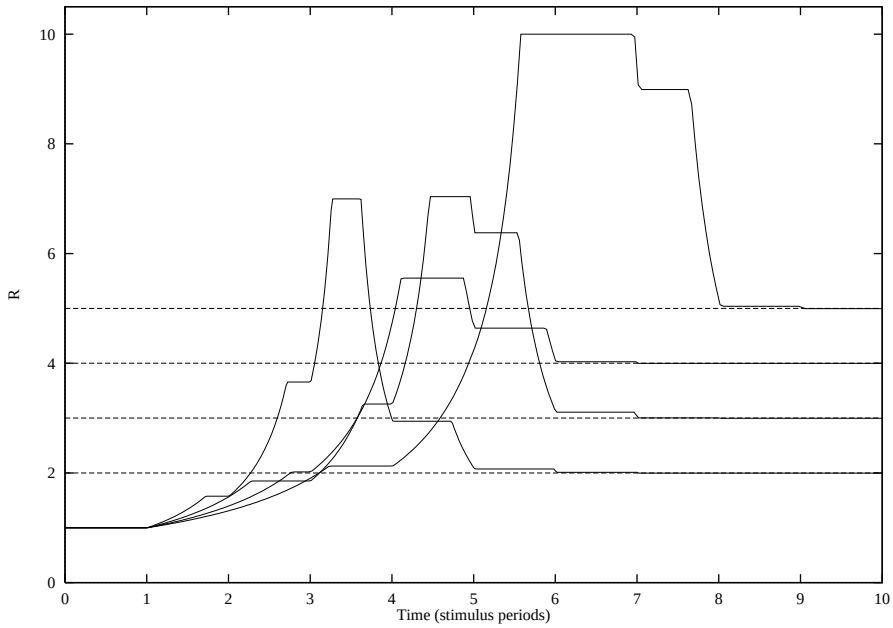


Figure 8.17: Variation in R for initial $R = 1$ and various target values, using freeze strategy and "recurrent" gradient term.

be noted that these recurrent connections are of the same form as those used in the networks presented in earlier chapters. It is of course possible to replace the squared activation term with a sigmoid function which behaves similarly in the relevant range. The modified circuit for the neural phase-locked loop is shown in Figure 8.18.

8.7 Tempo range, spontaneous tempo and preferred tempo

Many studies into the perception of musical rhythm note the existence of a tempo range characteristic of a number of different activities. Fraisse [59] discusses typical rates for spontaneous tapping, i.e. tapping in the absence of a stimulus with which to synchronize, and notes that the period varies between 380 ms and 880 ms. In this task the inter-subject variability is much greater than the within-subject variability (except, interestingly, in the case of identical twins). A distinction is made between the *spontaneous* tempo and the *preferred* tempo, the latter being defined as the tempo which a given subject perceives as being "most natural", that is neither to fast nor too slow. The preferred tempo has most often been found

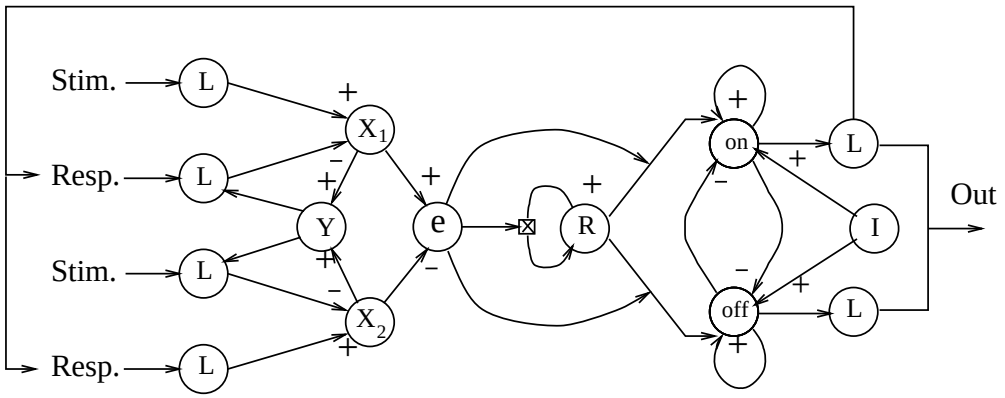


Figure 8.18: The modified neural PLL.

to have a period of 500-600 ms. Fraisse points out the similarities between the periods of the spontaneous and preferred tempo, as well as typical tempos for other activities including walking, rocking and heartbeat, but is careful not to infer a common pacemaker. Regarding the synchronization of activity (e.g. tapping) with an auditory stimulus, he notes that this phenomenon appears very early in life and is a tendency which adults find very hard to resist. The range of tempos for which synchronization of motor activity is most accurate is approximately 400-800 ms, once again centred on a value approximately equal to the preferred tempo, and synchronization is achieved "from the third sound on" (a level of performance matched by the circuit developed above). Synchronization is also possible for tempos outside of this range— Fraisse estimates limits of about 200-1800 ms. He further relates these limits to the perceptual limits upon which all rhythm perception is based: 200 ms is approximately the lower limit for the perception of two separate stimuli, whereas 1800 ms corresponds to the separation at which two events become perceptually disassociated, i.e. considered as individual events. One final point that Fraisse makes regarding synchronization of motor activity with an auditory stimulus concerns a difference in the performance depending on whether the hand or the foot is used. It has been found that the tapping of a forefinger *precedes* the stimulus by about 30 ms, but that this value is greater if the foot is tapped instead. Fraisse accounts for this by postulating that synchronicity is defined by the "coincidence of the auditory and

of the tactile-kinesthetic information at the cortical level" and that the observed effect can be explained by the longer transmission time for the latter information when the foot is tapped, as opposed to the finger. This gives some support to the approach used in this chapter, in which synchronization is defined as the phase-locking of two distinct signals as they arrive at part of the brain.

Other authors support Fraisse's conclusions, in particular with relation to musical stimuli. Povel [126] notes that beat periods in western music range from approximately 250 ms to 1500 ms, and that these beats induce a rhythmic response in the listener. Warren et. al. [155] determine that the recognition of melodic themes is limited to inter-onset-intervals in the approximate range 200-1200 ms, even though Subjects can discriminate different ordering of the same items at much higher rates — they suggest a link between typical melody rates and typical speaking rates. Handel and Oshinsky [74], in an series of experiments in which Subjects were asked to tap in response to different polyrhythms, note that the positions of taps relative to the polyrhythms changed with the tempo: for example, when presented with either a 2 by 3 polyrhythm or a 3 by 5 polyrhythm at various presentation rates, Subjects preferred to tap along with the quicker stimulus until its inter-element interval became less than approximately 200-250 ms, at which point they switched to the slower stimulus. A further increase in the presentation rate, sufficient to bring the inter-element interval of the slower stimulus close to the the same limit, led to an increased tendency to tap in synchrony with the points at which the two rhythms coincided. This pattern of responses was repeated for many different polyrhythms and indicates a definite lower limit on the period of the tapping response. The results also confirm the existence of an upper limit (in period) beyond which events are considered distinct. Handel and Oshinsky conclude that

> "at faster rates, the elements of a pulse train are perceived as grouped, with subjectively accented and unaccented elements, and therefore cannot serve directly as the meter. The subunits, however, can serve as the meter, and this leads to tapping every second, third or fourth element or tapping every pattern repetition".

This idea will be echoed later in the chapter in relation to the use of the pacemaker to model the perception of simple metres.

The existence of a spontaneous tempo and a preferred tempo, both in the region of a 600 ms period might suggest that in the absence of a stimulus signal, the pacemaker relaxes to a state

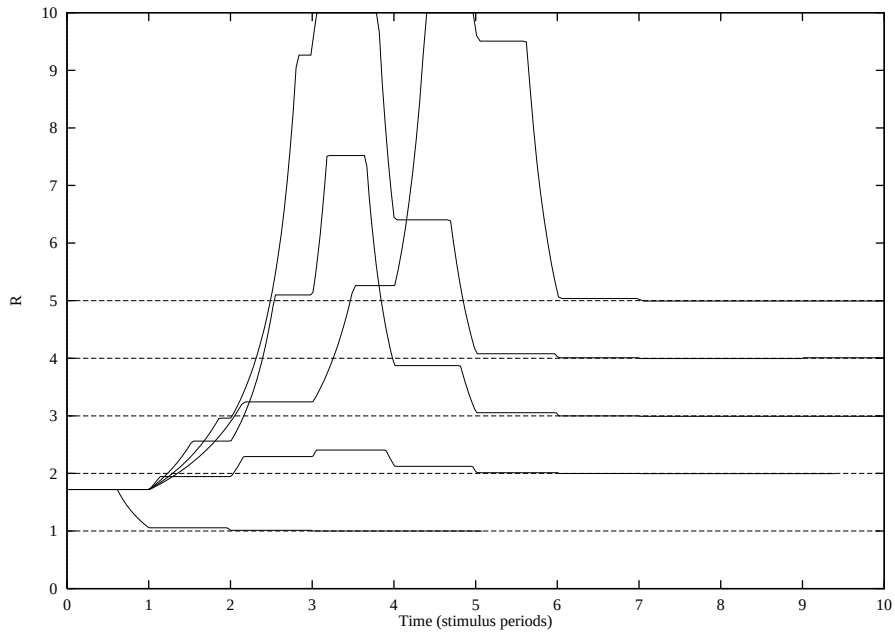


Figure 8.19: Variation in R from an initial "default" value to various target values.

in which the value of R is consistent with such a period. Because of the "freeze" strategy used during stimulus tracking, the circuit performs most efficiently when its initial value of R is higher than its target value. Figure 8.19 shows the performance of the network for various stimulus periods for the case in which the initial response period is set at approximately 600 ms. It can be seen that, for fast stimuli, synchronization can take up to seven stimulus periods (although the time taken to achieve phase lock is approximately constant). To allow for a "default" value of R consistent with the spontaneous tempo, but also permit fast tracking of a stimulus when necessary, we might postulate that the "instruction" to track a stimulus automatically pulls the value of R up to a level near to its upper limit, from where it can quickly relax to a level appropriate to the stimulus.

8.8 Coping with "missing" onsets

The simulations presented above assume the provision of a stimulus of fixed period with which the pacemaker output acquires phase-lock. It has been shown that a relatively simple tracking circuit is able to achieve a level of performance approaching that of human Subjects. One of the reasons for developing such a circuit, however, was to supply a rhythmic attentional

pulse which could be used in conjunction with the **SONNET** networks of earlier chapters, in particular allowing them to differentiate between the same set of notes presented in different rhythms. This clearly requires that the pacemaker produce a response pulse at times when an "expected" note onset is missing. For example, in order to distinguish the sequences $(3, 2, 1, 3, 2, 1)$ and $(3, 2, 1^2, 3, 2, 1^2)$ we require the pacemaker to provide a pulse at the midpoint of each of the longer notes. Unfortunately, use of the scheme developed so far leads to large variation in the value of the response period, since a response pulse occurring in this way, in the absence of a stimulus onset, will lead to the generation of an error signal, which will start to lower the value of R. The error signal will turn off at the onset of the next stimulus event (the "3" in the sequence above), at which point the response period will be approximately twice its previous value. The onset of the subsequent event ("2") will therefore cause the value of R to rise again towards its original value with overshoot inevitable. This instability in the value of R, when the circuit is presented with such a simple sequence, is clearly unsatisfactory—we would expect a listener faced with the same rhythmic sequence to continue to tap at a constant rate, unaffected by the different note lengths.

In order to avoid this problem, we might decide to "lock" the pacemaker by simply disabling the action of the error signal. Unfortunately, locking the pacemaker in this way leaves it insensitive to genuine changes in the tempo of the stimulus, changes which should ideally be tracked by varying the response period. It would be very difficult to establish a set of conditions under which locking of the pacemaker should occur, which would allow the pacemaker simultaneously to provide attentional pulses as required, to remain stable when presented with non-isochronous rhythms and to track genuine changes in tempo.

This problem has been noted by others who have sought to provide models of "beat tracking". Allen and Dannenberg [3] describe the problem as it relates to the beat-tracking method of Dannenberg and Mont-Reynaud [32]. This method employs two parameters, namely the *decay rate* and the *confidence*. In discussing the first of these, the authors touch upon the issue raised above: a high decay rate allows the tracker to vary its tempo rapidly in response to the timing of the most recent notes, but does so at the expense of stability; a low decay rate, however, provides tempo stability at the expense of a slow response rate. The confidence parameter weights the importance of onset events depending on the extent to which they can be considered to fall on "downbeats". The fact that the system uses this second parameter

to place particular emphasis on the position of *downbeats* is an idea carried forward to the solution presented below. Rosenthal [134] uses an "artificial intelligence" approach to emulate rhythm perception, which searches for regularly spaced onsets to define potential rhythmic levels, subsequently favouring those levels which can best be organized into hierarchical "families". His model assumes the existence of so-called "ghost events" which "represent missing notes—places in the music where one might tap one's foot, even though no note begins there". These ghost events are positioned so as to subdivide equally the period between events at the next "higher level".

The solution proposed here involves a differentiation between the rôles played by the output pulses generated by each of the pacemaker's component cells. In particular, we assume that *both* the on-cell and the off-cell can generate output pulses, but that the output pulses from *only one of the cells* (for instance, the on-cell) are forwarded to the phase-detector cells where comparison with the stimulus signal takes place—this is as it was shown in Figures 8.11 and 8.18. By contrast, the attentional signal, which is postulated to feed into the SONNET network, is taken as the combined output from *both* pacemaker cells—this combined signal is labelled "Out" in the relevant figures. The necessary additional assumption is that the stimulus signal does not represent the onset of *all* stimulus events but only the onsets of events that can be considered to be *accented*. To illustrate this idea, we shall once again use the sequence $(3, 2, 1^2, 3, 2, 1^2)$ as an example: we shall assume that the accents fall on each occurrence of the "3" and the "1". Faced with this sequence, the pacemaker synchronizes the output from, say, its on-cell with the accents, which are evenly spaced in time. The off-cell activity is 180 degrees out of phase with the on-cell and supplies attentional pulses at both onsets of the item "2", as well as at the mid-point of the longer notes—these off-cell pulses are not fed back to the phase-detector and therefore play no part in the tracking process.

The decision to use the onsets of *accented* notes to drive tempo tracking can be seen to be necessary when one considers the potential which unaccented notes (e.g. "grace" notes) would otherwise have for upsetting the tracking process. Other models of the process of rhythm perception make similar assumptions. As noted above, the model of Dannenberg and Mont-Reynaud [32] uses the location of "downbeats" to drive its tracking process, ignoring those events that occur "at points exactly between two adjacent downbeats". The hierarchical clock proposed by Povel and Essens [127] is established with regard to the position of accents:

the authors describe, with reference to Thomassen [148] and Povel and Ockermann [125], some of the conditions under which accents are perceived, even in sequences consisting of ostensibly identical tones. They conclude that "since perceptually these sequences do possess accented and non-accented events, it seems reasonable to assume that the occurrence—and more precisely, the distribution—of the accented events is the main determinant of the hypothesized internal clock". Other authors (Palmer and Krumhansl [121], Clarke [28], Jones [86], Jones and Boltz [87], etc.), also describe a variety of conditions under which events will be perceived as accented, regardless of whether a metre has been established. The fact that the perception of accents does not rely on the prior establishment of a "beat" allows us to avoid the potential circularity associated with the notion that the "beat" is itself established with reference to the location of accents. Of course, once a beat has been established then it may lead to the accentuation of notes which might otherwise have remained unaccented—this lends some conservatism to the tracking process. The model of Longuet-Higgins and Lee [108], though not making explicit mention of accents, makes the assumption that "relatively long notes...initiate major metrical units, since they are perceptually more salient than short notes": these more salient notes are instrumental in establishing the metre.

8.9 Alternative pacemakers

In the previous section we saw that it was possible to use the standard gated pacemaker model to track a sequence in which alternate events were accented, i.e. a sequence presented in a duple metre. Another possibility is that the sequence is presented with every third event accented—a triple metre. Clearly the model presented previously is not adequate to cope with such a sequence: if the output pulses from one of the pacemaker cells, say the on-cell, are locked in phase with the onsets of the accented events, then the pulses associated with the other cell will occur midway between the accents and will thus fail to coincide with either of the two intermediate non-accented events. To deal with this problem it is necessary to postulate the existence of a three-way gated pacemaker as shown in Figure 8.20. A simulation of the behaviour of this circuit, using the same set of parameters as was used for the two-way pacemaker, is shown in Figure 8.21. It should be noted that the value of R needed to drive the three-way pacemaker at a given *inter-accent* period differs from that required to drive the two-way pacemaker at the same *inter-accent* period. This is because the value of R controls

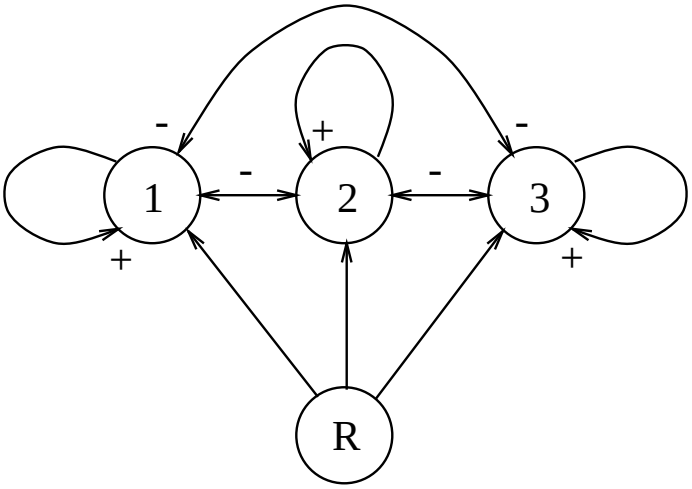


Figure 8.20: A 3-way pacemaker (arousal source not shown).

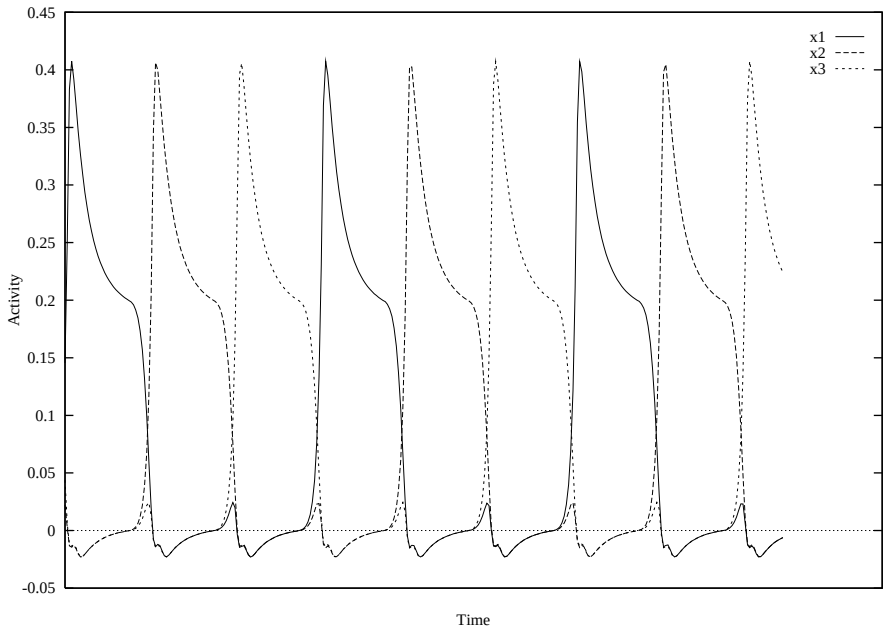


Figure 8.21: Activations in a 3-way pacemaker.

the time for which each of the component cells can in turn sustain a high activity, i.e. R controls the time period between pulses in the attentional output signal (i.e. the signal taken from *all* pacemaker cells). This time period is approximately constant, for a given value of R, regardless of the "dimensionality" of the pacemaker. A three-way pacemaker, in which *one* of the cells produces output pulses which align with stimulus accents, can produce an attentional signal appropriate to a sequence presented in triple metre.

The question arises as to how the attentional signal most appropriate to a given stimulus sequence might be established by the network. On presentation of the sequence, which we will assume is presented either in a duple metre (every second event accented) or in a triple metre (every third event accented), it is postulated that one of *each type* of pacemaker (binary and ternary) begins to track the stimulus by aligning the output of *one* of its constituent cells with the onset of accented events. Clearly a decision has to be made as to which of these two pacemakers is producing the attentional output (the summed output of all cells) most appropriate to the stimulus, since *both* have acquired phase-lock with the stimulus accents. This can be achieved by obtaining a measurement of the level of *correlation* that each of the attentional signals has with the signal derived from the onsets of *both accented and non-accented* stimulus events, or, more particularly, events that are considered accents at the immediately lower level. As noted previously, if this correlation were perfect then there would be no need to abstract an attentional signal from the (complete) stimulus signal at all, since the two would be identical. Although changes in the stimulus rhythm will thus serve to lower the correlation between even the "most appropriate" attentional signal and the stimulus, we can still expect that the correlation measure will allow the network to determine the *relative* appropriateness of the attentional signals produced by each type of pacemaker. A simple neural circuit for determining the approximate correlation of the relevant signals is shown in Figure 8.22: when neither input latch cell is active, the activity of the correlation cell, C, remains constant; when only one of the two signal cells is active C's activity is pulled down towards its minimum value (e.g. zero) by the action of a negative input; by contrast, when *both* signals are active, C's activity is pulled upwards towards its maximum value by the action of a positive input. The C cells from each of "candidate" pacemakers can feed their activities forward to a field of competing cells, at which a "choice" is made corresponding to the pacemaker with the greatest correlation value. The attentional signal emanating from the

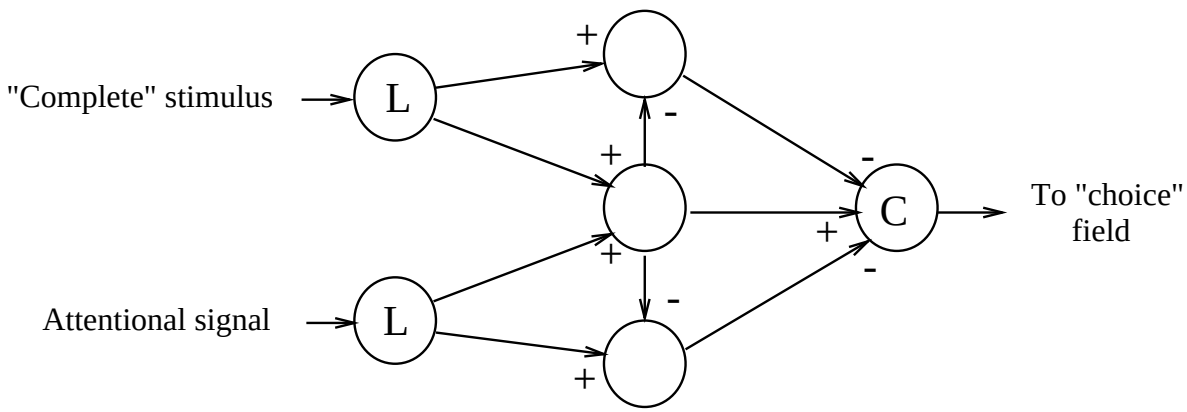


Figure 8.22: A circuit for assessing the correlation of two signals.

chosen pacemaker will then be enabled, others will be disabled. The activity of the cell C can be assumed to obey the standard differential equation for a neuron with additive dynamics, in which the decay term is set very low or at zero. Adjustment of the other parameters will determine the time constants with which the activity of C approaches its upper and its lower limits, allowing the correlation to be established over a period of perhaps one or two measures (inter-accent periods). Figure 8.23 shows a simple example of the correlation detector at work: the stimulus starts in duple metre with two pacemakers, one binary the other ternary, tracking its accents. The correlation value for the binary pacemaker, C_1 is initially maximal and dominates that of the ternary pacemaker, C_2. However, when the stimulus abruptly changes to triple metre, C_2 starts to rise as C_1 falls. After a short time (1-2 measures) C_2 dominates C_1, indicating that the ternary pacemaker is now producing the more appropriate attentional output. The "time lag" in the above process helps to smooth over the effect of any particular presentation rhythm, leading to a more stable assessment of the appropriate signal, and in turn will lead to an in-built conservatism, i.e. a tendency to persist with a given metrical assessment even in the face of a short period of contradictory information. This conservatism has been noted by many authors (Longuet-Higgins and Lee [108], Lee [103], Palmer and Krumhansl [122] etc.), as has the related tendency for composers to seek to establish a metre early in a piece of music, before allowing "artful violations" of the established metrical framework to occur. Support for the notion that listeners are able to choose between

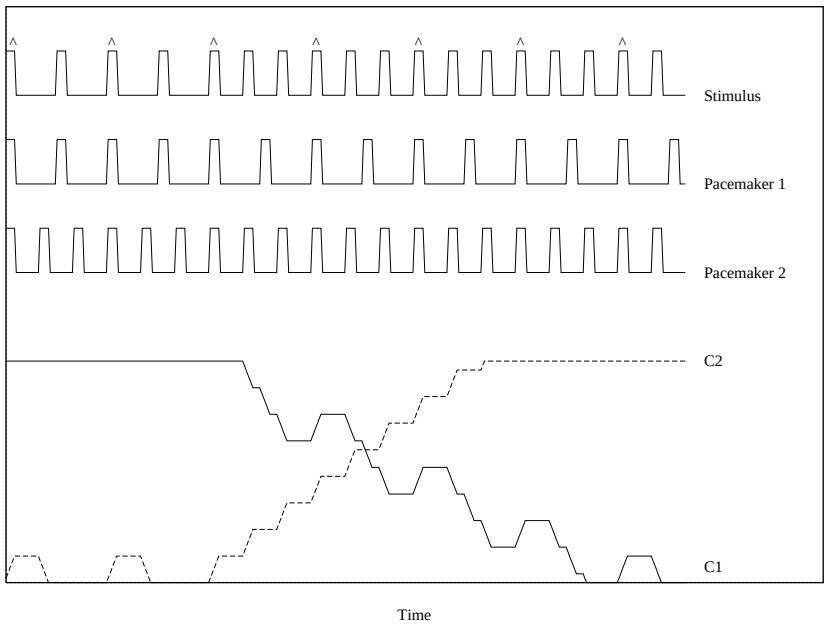


Figure 8.23: The effect of a change in stimulus metre on the C values of two pacemakers.

different metrical interpretations of the same stimulus by an appeal to the "relative frequencies of events' temporal locations", is provided by [122].

Having considered the cases in which stimuli are presented in either a simple duple metre or a simple triple metre, it is now necessary to consider the ways in which more complicated metrical frameworks might be perceived. The first possibility is a stimulus in which every *fourth* note is accented. My simulations have shown that it is possible to produce a four-way pacemaker that behaves in the same way as the binary and ternary pacemakers described earlier. This could certainly be used to generate an appropriate attentional signal in the manner detailed above. Although a five-way pacemaker was also simulated it seems unlikely that such a circuit would be put to common use, given the paucity of musical stimuli in which every fifth beat is accented. Indeed, with the framework developed here, we can see the correspondence between the most commonly used metres, i.e. those which accent every second, third and fourth beat, and the most structurally simple pacemakers.

A further development would be the provision of a "hierarchy" of pacemakers, which could be used to model a metre such as the 6/8 metre. In this case, it is necessary to postulate two levels of accent strength since, not only is every third beat accented, but also every second accent is itself accented. To model such a situation we must supply two (non-competing)

pacemakers, one binary pacemaker which uses the output from one of its cells to track the occurrence of "accented accents", the other a ternary pacemaker which operates at the lower level. Two questions arise regarding such an arrangement. Firstly, whether the ternary pacemaker tracks the occurrence of stimulus accents or, alternatively, tracks the attentional output pulses produced by the binary pacemaker. The latter option would seem to be the most efficient, since it would allow the perception of the meter to persist at points at which one of the weaker (non-accented) accents was missed out—in other words, by moving the level at which the pacemakers track the stimulus up to the level of accented accents we add to the ability of the combined pacemaker to cope with rhythmic variations at lower levels without introducing instability. At the same time, however, we should note that in doing so we decrease the speed with which the resulting circuit is able to track genuine changes in the tempo. Secondly, we should ask what circumstances allow two pacemakers to be linked in this way to generate a compound metre. Palmer and Krumhansl [122] present evidence that experienced musicians are more prone to abstract a metrical *hierarchy* from an impoverished stimulus than are untrained Subjects—the authors suggest that experienced musicians have "an abstract knowledge of multileveled metrical hierarchies". The performance differential between musically experienced and inexperienced Subjects suggests that *learning* plays a rôle in the ability to establish metrical hierarchies. Correspondingly, this suggests that the additional circuitry that allows two simple pacemakers to be yoked together to form a compound pacemaker also results from learning at a neural level. Further work is necessary to investigate possible mechanisms by which such learning might be accomplished.

8.10 The work of Jones et. al

In a large body of work, the psychologist M.R.Jones has developed models which highlight the rôle of rhythmic or "dynamic" attending in the perception of music. The ideas which she proposes relate closely to some of the work described in this chapter and provide a theoretical framework within which this work can be further developed. In order to illustrate these ideas I shall detail briefly those presented in two recent papers, viz. Jones [86] and Jones and Boltz [87].

Jones [86] begins by detailing some elementary ways in which the melodic and rhythmic aspects can be separately described in relation to monophonic music, before attempting to

unite them into a more integrated framework. With regard to pitch relationships she identifies three types of melodic accent: contour-based accents, pitch-interval accents and tonal accents. Similarly, with regard to time-relationships she refers both to the tempo, as described by the beat period, and to the metre, in which a small number of beat periods are subsumed into a larger period, the measure. Rhythmic accents are described with reference to the *serial durational patterns* formed by the melodic stimulus. In order to integrate the pitch and time relationships Jones then introduces the concept of the "joint accent structure", which results from the combination of the melodic and rhythmic accents. At the heart of this concept lies the assumption that accent couplings, that is points at which melodic and rhythmic accents co-occur, can introduce variation in accent strength, and furthermore that the strong accents thus defined tend to be separated by a constant time span (number of beats). An inverse relationship between accent strength and time period is noted, as is the fact that the time periods at adjacent levels in the joint accent structure are often related by small-integer "time span ratios". When adjacent levels are so related the joint-accent structure time hierarchy is said to be coherent. Jones cites Essens [55] and Boltz and Jones [16] in support of the idea that coherence in the joint-accent structure time hierarchy aids in the performance of reproduction tasks. Her thesis, that coherence in the time hierarchy (derived as it is from the locations of accents of varying strength) allows attentional resources to be properly directed towards points in time at which events are likely to occur, overlaps considerably with the ideas presented in this chapter.

This idea is developed further by Jones and Boltz [87], who postulate the existence of two modes of dynamic attending, namely *future-oriented attending* and *analytic attending*. They suggest that future-oriented attending is primarily employed for stimuli of high coherence, whereas analytic attending is most applicable to low coherence stimuli. Once again they make it clear that the concept of coherence refers to the hierarchical (nested) arrangement of time periods defined by the regular distribution of objective (sic) accents of varying relative strengths and of varying sources (both temporal and non-temporal). Simple time hierarchies are assumed to be those based on small integer time-ratios (typically 2 or 3) relating adjacent levels.

Crucial to the idea of dynamic attending is the concept of *attunement* whereby "one or more attending rhythms become phase-locked to corresponding time spans marked within

the event ... [which is] most apt to occur with temporally coherent events". Although several levels may exist in the hierarchy, corresponding to different time periods and accent strengths, Jones and Boltz suggest that one of these levels can be denoted the referent level. The referent level approximately corresponds to the music-theoretical concept of the "tactus". They suggest that an attender, in particular a *skilled* attender, can vary the level at which they are attending (the so-called focal level) by a process of attunement shifting. In the model developed in this chapter, this would correspond to focussing on the output generated by a particular pacemaker, assuming that different pacemakers had acquired phase-lock with accents at different levels. It was noted earlier that experienced musicians were prone to infer metrical hierarchies of greater "depth" than less experienced Subjects. Jones and Boltz suggest that their skill lies in the ability to shift easily between levels of a hierarchy already established. In practice the ability of the attender to shift focal levels is restricted by the limits of the so-called "dominance region" which extends only a small number of levels either side of the referent level. Their suggestion that attunement shifts "allow one to selectively "examine" strings of notes marking a given hierarchical level" corresponds precisely with one of the major functions of the rhythmic attentional signal as it affects the sequential memory described in earlier chapters. Furthermore, the idea that future-oriented attending operates for coherent stimuli, particularly at levels at or above the referent level, gives justification for the assumption that it is the events at such levels that cue expectancies regarding likely future events.

In summary, the work of Jones, particularly as characterized in the two papers described above, offers considerable support to many of the ideas developed in this chapter.

8.11 Tempo-invariant durational encoding

The final section of this chapter is concerned with the way in which rhythmic or durational patterns can be represented. In particular, it will be shown how the pacemaker developed above can be used to encode durations in a tempo-invariant manner. Tempo invariance, operative between certain limits, is required of any system that attempts to model the perception of event durations in a piece of music—the fact that event durations are specified in terms of "numbers of beats", rather than in terms of absolute time periods, indicates this strongly. Throughout this section the duration of an event will be taken to be equal to the inter-

onset-interval defined by the event and its successor. Such a definition avoids complications introduced by variations in the staccato/legato dimension.

8.11.1 Grossberg's "rhythm generator"

Grossberg [69] discusses a rhythm generator which is able to extract duration information from an event sequence in such a way that the sequence can subsequently be performed correctly (i.e the correct items in the correct rhythm) out of short-term memory. The basic idea behind the rhythm generator is that, as each event occurs, it not only sends a signal to the STM field representing item-order information, but also sends another so-called "arousal pulse" to a field storing rhythm information. The arousal pulse remains active for the entire duration of the event and processing at the rhythm field converts this duration into a "topographically organized STM *intensity*". It is not obvious precisely what is meant by this phrase, but it is clear that for each event in the sequence a corresponding cell in the rhythm field acquires an activation proportional to the duration of the event—once again the mechanism by which the event duration is converted into a proportionate cell activity is not outlined. Another aspect of the network which is left unexplained is the way in which cells in the item field, which store item order information in the pattern of their relative activation, are associated with cells in the rhythm field. By considering the means by which items can be read out of short term memory both in the correct order and with the correct durations, we can see that there must be some association between those cells in the two fields whose activities represent information concerning the same event. This is clearly necessary, since the activities of the cells in the rhythm field represent *durations* of events and therefore cannot also store information pertaining to the order in which those durations occurred. Accordingly we must assume that each cell in the rhythm field is associated with a cell in the item field, and that the order in which a series of *durations* is read out of STM is specified by the time ordering of *the associated items*. The simplest scheme of this type would be one in which each item cell has a "companion" cell which stores the duration information relevant to that item. All such schemes, however, have a major disadvantage, namely that the representation of the rhythm is "tied" to the items that defined the rhythm—this problem is also noted by Nigrin [117]. To illustrate the point, consider the case in which a rhythm is defined by the onsets of the repeated item A. It is not easy to see how the type of scheme suggested above would cope with

a request to repeat a different item, for instance item B, in the same rhythm, even though such a task is trivially accomplished in vivo. Moreover, a rhythm defined by a collection of unfamiliar and wildly different sounds can be easily recalled from short-term memory. It seems that the rhythm in which a sequence of items is presented can be "factorized out" of the stimulus and be recalled independently of the items that defined it. Such considerations cast doubt on the model apparently proposed by Grossberg. Furthermore, there is no indication as to how tempo invariance might be introduced into the process of durational encoding, though this was never claimed as a feature of the model.

8.11.2 An alternative scheme

Two ideas underlie the development of my alternative scheme, similar to Grossberg's but with additional features which allow it to overcome the problems discussed in the previous section. Firstly, that the process of converting an event duration into a proportional cell activity can be modulated by a "tempo signal" taken from an appropriate pacemaker. Secondly that the cell activity thus derived can further be converted into a *position* in a "spatial map" (defined by Grossberg and Kuperstein [72] as "a network which encodes information by activating cells in different locations"). A simple circuit which satisfies both these requirements is shown in Figure 8.24. The input to a cell, whose activity and identity will be denoted x, is taken from the rate-determining cell of a pacemaker which is tracking the onsets of stimulus accents. Changes in the activity of x can be described by the additive equation

$$\frac{d}{dt}x = I - KE \tag{8.16}$$

where I denotes the input to cell x, such that $I = f(R)$, K is a large constant, $E = 1$ for a short time after the occurrence of an event and $E = 0$ otherwise. This part of the circuit serves to convert the duration of an event into a proportionate activity, x, in the following way. When an event occurs the value of E is set to one for a short time during which the activity x drops very rapidly to its minimum value (assumed to be zero) under the influence of the term $-KE$ in Equation 8.16. When E falls to zero, however, the value of activity x starts to rise at a rate set by the value of $f(R)$, which we shall assume to be constant throughout the event's duration. Momentarily before the onset of the next event, therefore, the maximum value reached by the activity x will be very nearly proportional to the inter-onset interval defined

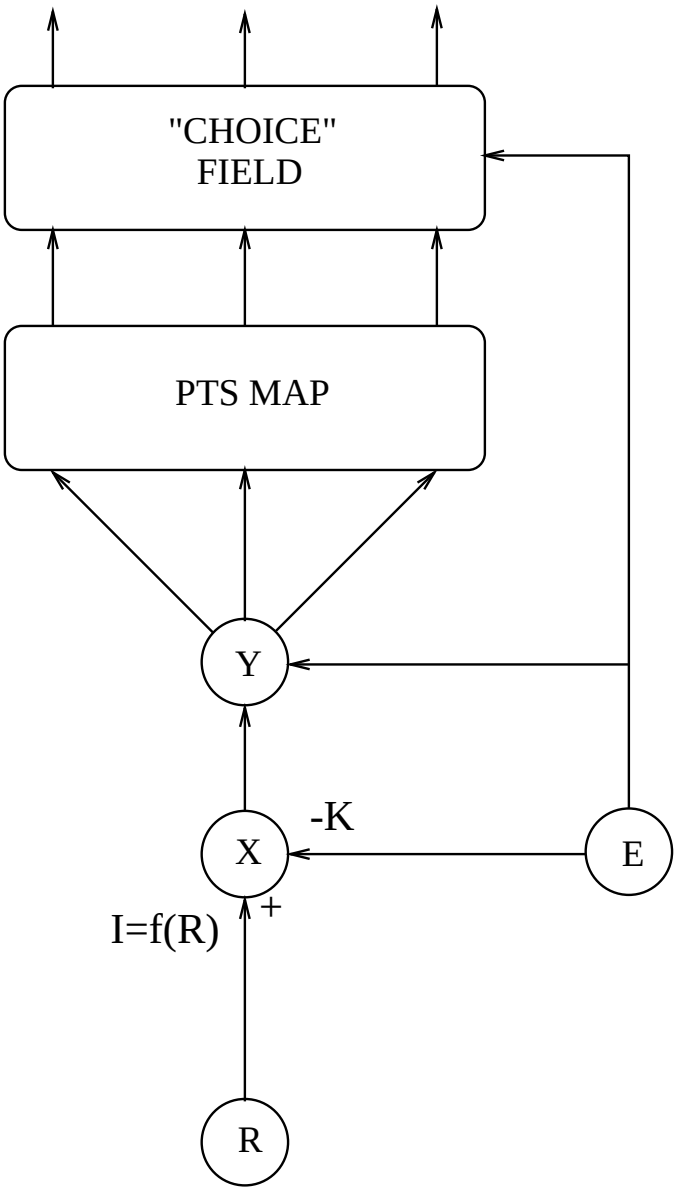


Figure 8.24: The suggested circuit for a "rhythm generator".

by the two successive events. In practice we add a further cell, y, whose activity rapidly tracks the value of x, keeping $x = y$ at all times when $E = 0$; when $E = 1$ the activity y is held constant. This modification allows the value that x has reached just before the onset of the next event, to be "held" briefly at cell y, whilst x drops rapidly to zero under the influence of the large negative input, $-KE$. Thus the first circuit requirement, that an event duration is converted into a proportionate cell activation, y, is satisfied. Note also that, under certain conditions placed on the input to cell x, this conversion is done in a tempo-invariant manner. These conditions require that a plot of the value of $f(R)$ against the frequency of input events (measured in, say, beats per minute) approximates, within the appropriate frequency range, a straight line passing through the origin. If this is the case then a doubling of the tempo will result in a doubling of the value of $f(R)$. To illustrate the tempo invariance, suppose that an event's duration is defined as being equal to one beat's duration. At a given tempo, T_1, the absolute period of one beat is denoted p_1 and the corresponding value of R (the activity of the rate-determining cell of a pacemaker that is tracking the stimulus) is denoted R_1. Given these values we can calculate that the activity reached by the cell, y, will be approximately equal to $f(R_1).p_1$. If we now double the tempo so that the new tempo, $T_2 = 2T_1$, then we also half the absolute time equal to one beat's duration, i.e. $p_2 = p_1/2$. Remembering that a doubling of the tempo results in a doubling of the value of $f(R)$, we have that $f(R_2) = 2f(R_1)$. Thus, for the faster tempo, the maximum activity reached by the cell y in one beat's duration is equal to $f(R_2).p_2$, which is easily seen to be the same activity reached in the slower tempo case. In this way, by using $f(R)$ as a gauge of the tempo, we can ensure that a duration consisting of a *specified number of beats* is represented by the same activity, y, over a wide range of tempos. Finally, the specification of the function f is made simple by the fact that the plot of R versus stimulus frequency (see Figure 8.5) is approximately linear for a given range of values—all that is required is the addition of a small positive constant to make sure that the plot passes through the origin. In Figure 8.25, pacemaker frequency is plotted against $f(R)$, where $f(R) = (R + 0.3)$, together with a number of points illustrating the actual pacemaker performance. The function itself is simply a standard threshold-linear function with a small negative threshold.

The second part of the rhythm generator circuit involves the feeding forward of the activation y to a "Position-Threshold-Slope (PTS) map" ([72]). This consists of a field of cells

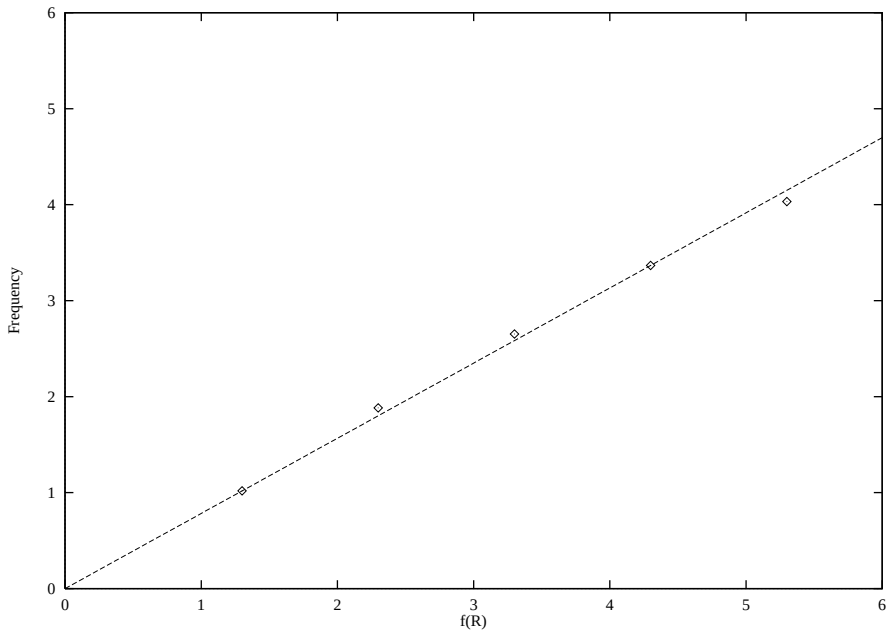


Figure 8.25: Comparison of actual pacemaker performance and linear approximation.

for each of which the output is linear function of the supra-threshold input, that is

$$O_i = S_i(I_i - \theta_i)^+ \tag{8.17}$$

where O_i denotes the output of the i^{th} cell, I_i denotes its input, θ_i denotes its threshold and S_i denotes the (constant) slope of the supra-threshold linear portion of its input-output function. The notation $(\gamma)^+$ is shorthand for $\max(\gamma, 0)$. The important point about a PTS map is that, if we assume the cells are arranged in a positional array indexed by the subscript i, then both the threshold, θ_i, and the slope, S_i, covary with index i, i.e. cells at a position in the field indexed by a larger value of i will have a larger threshold and a larger "sensitivity" to supra-threshold input. If one examines the array of outputs generated by the cells of a PTS map for various values of an input signal which is common to them all (i.e. I_i is the same for all i) we see that, as the value of the input increases, the locus of the cell with maximal output moves towards cells indexed by larger i. To illustrate this, assume that the covariance of threshold and slope with position i $(0 < i < N)$ is described by the following equations

$$\theta_i = i \tag{8.18}$$

$$S_i = K(2^i) \tag{8.19}$$

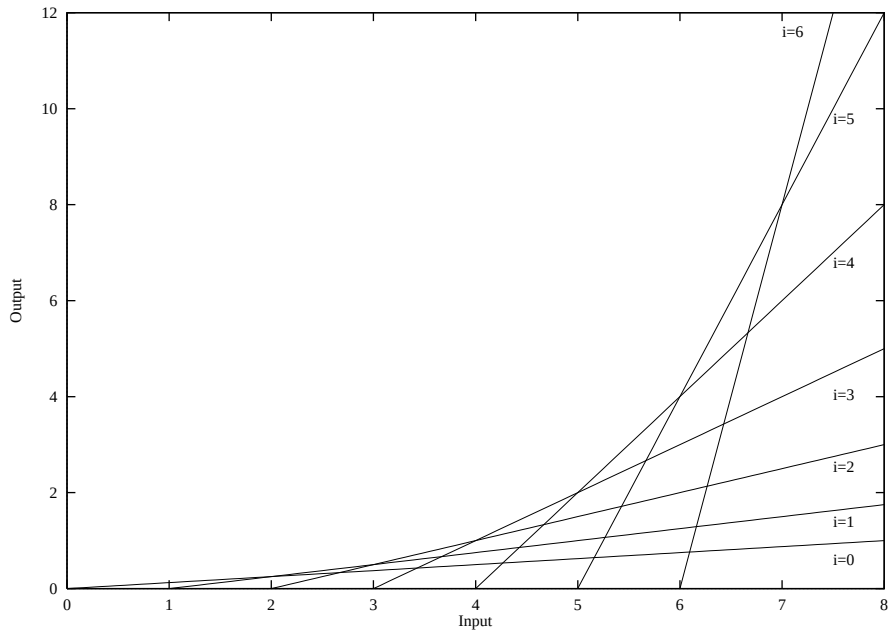


Figure 8.26: Input-output functions for various cells in a PTS map, indexed by i.

where K is a constant. The input-output functions for cells with index i varying between $i = 0$ and $i = 6$ and with $K = 0.125$ are shown in Figure 8.26. It can be seen that as the value of the input changes so does the index of the cell with maximal output—for each cell in the PTS map there is a range of input values over which its output exceeds that of all other cells in the field. This observation is the key to the second part of the rhythm generator circuit: we assume that the activity y, which represents an event duration ("measured" in beats), is fed forward equally to all cells in the PTS map. As the value of y increases, the locus of the cell with maximal output will move, as noted above, towards cells with large index i. When the next event occurs, the value of y will be briefly frozen, as will the pattern representing the outputs of the cells of the PTS field. At this moment (whilst $E = 1$) the PTS map output is fed forward (in a one-to-one mapping) to a field of cells configured in a competitive-cooperative (on-center off-surround) fashion. This field (the same as the "category field" in an **ART** network) quickly equilibrates so as to "choose" the cell with the greatest input, i.e. the cell that corresponds to the PTS cell with the maximum output. When the chosen cell reaches a certain threshold of activity, it forwards activity to the input field of, for instance, a SONNET module, and resets itself to zero. In this way the duration of the event is converted from the single cell activity, y, into a "focussed" pattern of activity across a field of cells. The

activation of one of these cells can be thought as the occurrence of a "durational item".

It should be noted that, depending on the parameters chosen for the PTS map, the activation of a given durational item could represent the occurrence of one of a range of input durations. Referring to Figure 8.26, we can see that, for a given index i, the i^{th} cell is "dominant" for a range of input activities (in this case representing durations). This can result in the "cleaning up" or quantization of an input sequence, where small variations in the precise item durations can be discounted. Essentially, each cell in the "choice" field, that contrast-enhances the output from the PTS map, acts as a one-dimensional "attractor"—further work is required to ensure that these attractors are centred at points representing common durations, such as those consisting of an integral number of beats (or even an integral number of half beats). This might result from learning in the connections from the PTS map to the choice field, which would alter the size of a given attractor's basin. The geography of the set of attractor basins could also be affected by the provision of top-down (expectation) input to the choice field. This would lead, for instance, to the perception of a given durational item, when that item was highly expected, even though the stimulus on its own might suggest a slightly different interpretation—further work is required to investigate these ideas. The model introduced by Desain and Honing [34], which is influenced by the connectionist paradigm, also seeks to model the quantization and categorical perception of durational patterns, albeit in a very different way. Desain [33] further employs this quantization model in a theory of rhythmic perception that subsumes a model of expectancy. Unlike the work presented here, the focus of this work is on modelling schematic expectancies, rather then the veridical rhythmic expectancies associated with a particular melody.

8.11.3 The chunking of durational patterns

As noted above, the conversion of a (tempo-invariant) duration into the focussed spatial pattern that results from contrast enhancement (choice) at a competitive-cooperative field, allows durations to be treated as items and, accordingly, to be presented to the lower levels of a hierarchy of **SONNET** modules. In this way, durational patterns can be stored in a sequential memory via the characteristic chunking of the sequence into what might be thought of as metrical feet, i.e. commonly occurring rhythmic phrases. This is consistent with the models of Mont-Reynaud and Goldstein [112] and Rosenthal [133], who "parse" rhythmic

sequences by chunking into commonly occurring subpatterns. It is also consistent with the model suggested by Vorberg and Hambuch [152], in which the "motor program by which appropriately ordered behaviour is generated" is considered to be structured hierarchically, whereas the execution of that program is considered to be mediated by a sequential timing mechanism. Pitch and durational information relating to a melody can thus be chunked and stored in parallel **SONNET** hierarchies. Ultimately, the presentation of the melody will result in the activation of a single high-level cell for each of the pitch and durational sequences. It is not unreasonable to assume that these two high-level cells, one responding to the pitch content of a given melody, the other responding to its durational content, would be associated at this level, and further associated with other cells representing, for instance, semantic information relevant to the melody's title or lyric. In this way, presentation of pitch information without duration information (e.g. isochronous presentation), should be able to cue the associated rhythmic information and, more particularly, the semantic information—this is consistent with the findings of those experiments cited in Chapter 2 (White [156], etc.). Likewise, durational information should be able to cue the pitch and semantic information, though with a level of performance restricted by the narrower range of likely input items—this is also supported by the experimental evidence cited in Chapter 2. Finally, the naming of a familiar melody should be able to cue the generation of both pitch and durational information resulting in the accurate recall (or "read-out") of the melody itself. A mechanism for this latter process is suggested in the next section.

This parallel synthesis of pitch and rhythmic information is consistent with the findings of Palmer and Krumhansl [121], who found that pitch and duration information made independent contributions to the perception of musical phrases. They also found that listeners familiar with the piece of music used in the study showed a higher correlation between pitch and temporal condition ratings than did Subjects unfamiliar with the piece. Their explanation, namely that familiar Subjects were able to imagine the pitch information when cued with the rhythmic information alone (and vice versa), is also compatible with the model described above, in which familiarity with a specific piece of music can lead to a high-level association between cells representing its pitch and durational content. The "imagery" suggested by Palmer and Krumhansl is thus mediated by both bottom-up and top-down processes.

8.11.4 Reading a melody out of long-term memory

I now propose a mechanism by which a melody, whose pitch and rhythm information has been stored in parallel, but associated, sequential memories, can be read out of long-term memory. The network relating to the read-out of rhythmic information is shown in Figure 8.27. The "bottom-up" section is the same as that shown in Figure 8.24, with its output feeding forward to the bottom level of a **SONNET** sequential memory. This **SONNET** memory and the parallel one associated with pitch information are assumed to be able to generate graded expectations for future duration and pitch items respectively, resulting from the top-down priming associated with, for instance, the naming of a melody. These expectations are represented by a pattern of activity at the bottom level of each **SONNET** hierarchy. This pattern can be subjected to non-specific arousal that results from a performance signal, P, that amplifies the cell activations in parallel until one of them (i.e. the one corresponding to the most expected item) reaches a performance threshold. At this point the relevant cell generates an output pulse and rapidly resets itself to zero. In the case of the pitch item, this output pulse is forwarded to a motor field that performs the relevant pitch event, whilst in the case of the duration item, the output pulse is fed, in a one-to-one fashion to the field labelled M in Figure 8.27. This field is a choice field which activates, during performance, that cell which most recently received an output pulse from the field above. The field labelled N receives additive input, also in a one-to-one fashion, from fields M and C. Since both fields supplying input to field N are choice fields, the pattern of activation at field N is assumed to be focussed, i.e. one cell will have maximal activity (assumed to be equal to one) and the others will be inactive. Activation at the field N in turn feeds down, via *weighted* connections, to the cell labelled z. The weights, w_i at these connections are assumed to have been learned via the equation

$$\frac{d}{dt}w_i = \epsilon E n_i (y - w_i) \tag{8.20}$$

where n_i is the activation of the i^{th} cell in field N, ϵ is the learning rate and E and y are as defined previously. This learning equation allows the weight from the i^{th} cell in field N to come to equal the (mean) activity y that excites the i^{th} cell in field C. The activation of the cell z is equal to its input i.e $z = \sum_i n_i w_i - y$. Finally, the performance signal, P, is assumed to be inhibited for all times at which $z > 0$.

The mechanism works as follows. We assume that, before the performance starts, pitch

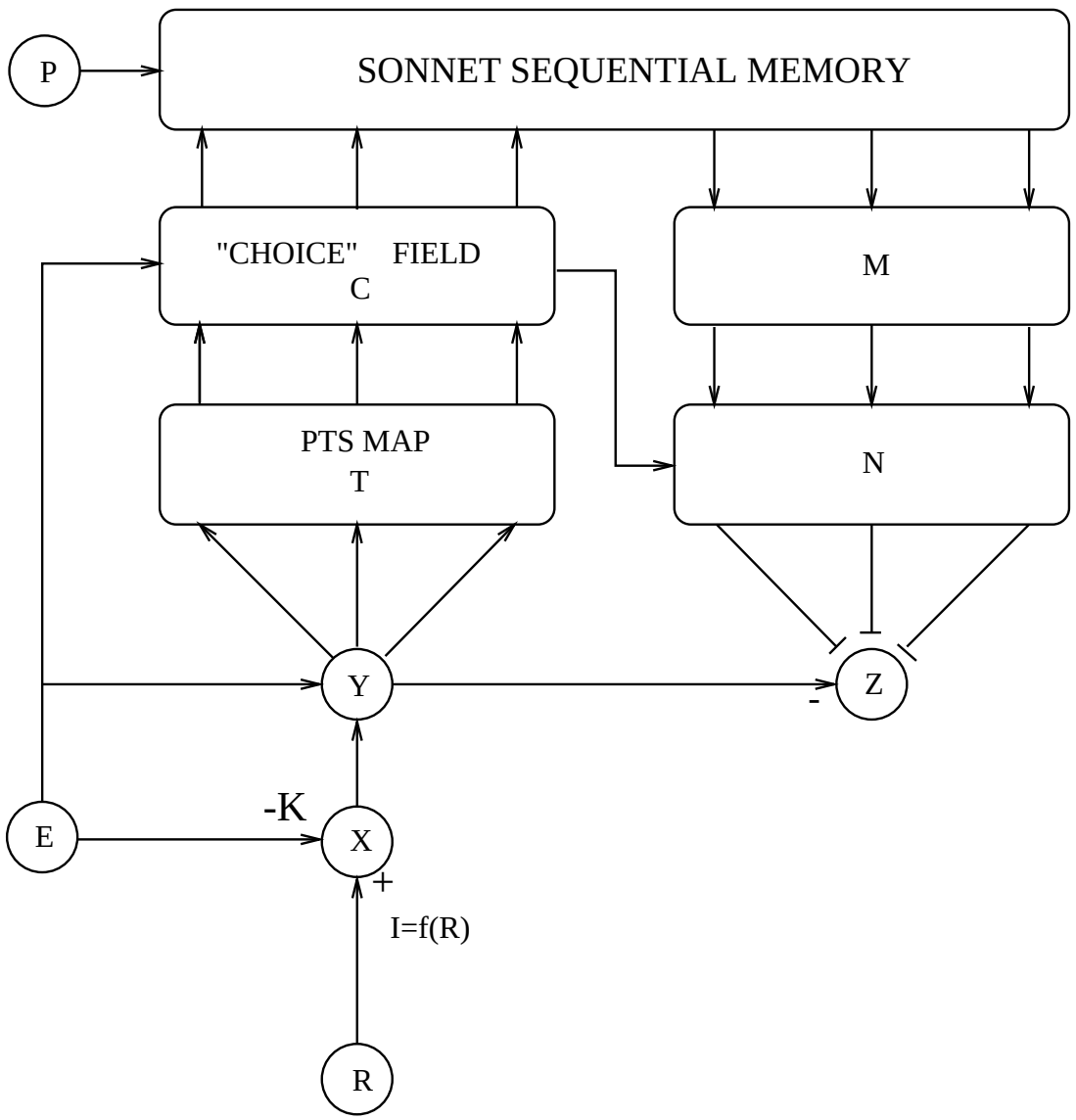


Figure 8.27: Mechanism for LTM read-out.

and duration expectations are present at the lower levels of the **SONNET** hierarchy due to top-down priming resulting from, say, the naming of a melody. Activities x and y are zero, as are all activities in the fields M and N. When the performance signal, P, is turned on, the most expected pitch item is performed whilst the most expected duration item is read into field M, and from there into field N. Top-down activity from field N excites cell z, setting its activation to a value proportional to the expected duration. Since z is positive, the performance signal is rapidly inhibited. Meanwhile, after a brief "event-pulse" from cell E, the activities x and y begin to grow under the influence of the input $f(R)$—the value of R is assumed to be set by "an act of will" so as to specify the performance tempo. The performance signal remains inhibited until the activity y grows sufficiently to take the z to zero—the time taken for this to occur depends on the expected duration of the current event. As soon as the performance signal becomes uninhibited, the next pitch item is read-out of memory and performed, causing an event pulse which drags activity x to zero. The "item" generated by the previous duration is forwarded up to the SONNET module and the next duration is read into fields M and N. The performance signal is again inhibited and the cycle begins again. At any time the tempo of the performance can be moderated by adjustment of the activity R.

8.12 Summary

In this chapter, I have introduced a "neural" circuit which is capable of tracking a rhythmic stimulus. Whilst no claim is made that this exact circuit exists *in vivo*, it illustrates that we can generate quite complex functionality from a small number of simple, "standard" neurons. The way in which the pacemaker can interact with the **SONNET** sequential memory was detailed and a compound model was introduced, which allows sequences of "durational items" to be stored, for both recognition and recall, in a tempo-invariant manner. The circuit is hard-wired at many points and further work is needed to investigate the points at which learning could usefully be introduced.

Chapter 9

Conclusions

9.1 Summary

There has been little previous research investigating the behaviour of masking-field-type networks. They are unusual, in particular because of the inhomogeneity in cell sizes and their restricted connectivity. The work presented in Chapter 4 and Chapter 5 highlights the inadequacies of the Cohen-Grossberg masking field, whilst the work in Chapter 6 shows how the **SONNET 1** network addresses many of these inadequacies. The modifications to **SONNET 1** suggested in that chapter resulted from extensive simulation and help to make it more appropriate to the learning of specific sequences and to the resultant generation of plausible expectations. The extensions to **SONNET 1**, described and simulated in Chapter 7, afford the network considerable power, and this is illustrated with respect to the learning of specific extended sequences. To my knowledge, this is the first application of a **SONNET**-type network to a real-world problem. The work presented in Chapter 8 suggests a model for dealing with rhythmic aspects of sequences, which *complements* the **SONNET**-type networks and can be combined with them in the processing of rhythmic patterns.

The connectionist models investigated and developed in this thesis, offer an attractive alternative to the other models of connectionist sequence-learning described in Chapter 3. Many of the latter, particularly those applied to music sequences, were variations of approach taken by Jordan [88] and Elman [54]. A critique of such models *qua* models of *perception* was given in Section 3.2.2. The network model developed here addresses each of the suggested criticisms:

- It is entirely self-organizing - a network's behaviour is described by a set of differential equations, which allow it to respond continuously to an environment in such a way as to learn regularities in that environment without any "external" interference in its operation. There is no distinction equivalent to that between the "training" and "testing" phases of alternative networks. The occurrence of an event leads to an immediate change in the network's state — there are no arbitrary time-delays such as are required in the other models. The network is able to run in "virtual time".

- It solves the stability-plasticity dilemma, so that the network can continue to learn new input patterns without any danger of eradicating knowledge previously acquired. This is of great practical importance, and is a prerequisite for any model which is to perform "on-line" learning.

- The stability of the network's learning is maintained even at high values of the learning-rate. This allows sequences to be learned in a relatively small number of presentations — this was not the case with the alternative models, which required many hundreds of learning "epochs" to learn a simple, short sequence.

- It is capable of modelling both recognition and recall of a given familiar sequence. Sequence recognition is characterized by the "bottom-up" activation of a single high-level cell unique to that sequence. It is postulated that associations between cells at this high level can allow the unification of different aspects of a given sequence e.g. pitch, rhythm, verbal accompaniment etc.. Such a mechanism would also permit the labelling (i.e. naming) of a stimulus sequence. Sequence recall, possibly cued by naming or partial presentation of a sequence, is accomplished by the top-down priming of low-level cells combined with a non-specific performance signal. A detailed description of a possible mechanism for recalling items in the appropriate rhythm is given in Chapter 8.

- The expectations generated by the network result from a context which can vary in size, depending on the way in which the input sequence is chunked into "phrases". In addition, expectations are evident at different hierarchical levels — we can have closure (i.e. zero expectation) at the note level whilst having expectations at the phrase level etc..

- The network deals naturally with sequences which possess a hierarchical structure. In fact, in order to learn an extended sequence, the network *must* impose a hierarchical chunking-structure. The inability to learn an extended sequence without breaking it into chunks (and "chunks of chunks" etc.) is a consequence of the limited capacity of short-term memory. This chunking strategy for sequence learning is consistent with much of the literature reviewed in Chapter 2 (see below).

- Once a phrase is learned in one context, it can be recognized in any number of different contexts. In the alternative models, the addition of "plan units" served to differentiate between the input patterns resulting from the same phrase in different contexts.

- Finally, all the inter-cell interactions postulated in the model rely solely on information available "locally" — the cells themselves are simple and their behaviour can be modelled with differential equations corresponding to "standard" neural models. All weight-learning is pseudo-Hebbian and likewise employs only locally available information. In addition, learning is unsupervised, unlike in many of the alternative models, which employ supervised-learning algorithms

As noted above, the **SONNET**-type model is consistent with much of the literature reviewed in Chapter 2. It is, in particular, consistent with the observation that melodies are remembered, in the *long-term*, as sequences of specific items (either scale-notes or intervals), rather than, for instance, as a contour "hung" on a scale. The grouping of melodies into *familiar* chunks offers an alternative to the "rule-systems" proposed by Restle, Restle and Brown, Deutsch and Feroe, Jones, etc., and allows us to unify the findings of experiments employing pitch-sequences with experiments employing stimuli less suited to the rule-based approach, e.g. letter-sequences. The effect of phrasing on the perception of sequences is also modelled well with the **SONNET**-type network: the placement of a "pause" within an otherwise familiar phrase will disrupt the recognition (and subsequent chunking-out) of that phrase; the placement of a pause between phrases will have no such deleterious effect, and will in fact aid in the chunking process. The experiments of Dowling [50] and Tan et al. [146], demonstrating poor recall for across-phrase subsequences, can also be related to chunking-out: if a phrase is chunked out shortly after its completion, then the subsequence formed by its latter notes and the first notes of the next phrase, will never find itself represented in working memory

and will therefore be difficult to recall. The correspondence between the **SONNET**-based model of melodic processing and Oura's theoretical model [119] is obvious, though further work is necessary to suggest a means of implementing the "modifiers" postulated by the latter model. The work of Carlsen, Schmuckler, Abe and Hoshino etc, relating melodic and harmonic expectations to culturally acquired "schemata", is also consistent with the current model. Bregman's [18] description of the rôle of sequential schemata is particularly apt:

> "Each schema incorporates information about one particular regularity in our environment ... we think that schemas become active when they detect, in the incoming sense data, the particular pattern they deal with. Because many of the patterns schemas look for extend over time, when part of the evidence is present and the schema is activated, it can prepare the perceptual process for the remainder of the pattern".

9.2 Further work

There is much work that can be done in continuing the research presented here. Firstly, the operation of the **SONNET**-type networks can be further refined: the current network model is complex and perhaps, in places, overly "mathematical", for example in its calculation of the learning moderator r_{ji} — it may be possible to "condense" the network somewhat. Paradoxically, the process of "simplification" may be achieved by the introduction of additional concepts: Nigrin [117] explores some possibilities in Chapter 8 of his thesis. One of these ideas appears particularly promising, namely the introduction of cells with oscillatory activation. Oscillations provide a mechanism by which the activities at distinct cells can be associated: coherent oscillations (i.e. oscillations which are in phase with each other) might, for instance, indicate predicates associated with the same input object. The potential for using synchronized oscillations in this way is discussed further in [151] and [73]. It is possible that this idea can be extended to encompass ideas relating to the attentional "streaming" described in Section 2.3.2, by postulating that the activities of cells corresponding to events in the same stream, should be synchronized in their oscillations. The difficulty found in "paying attention" to two separate streams simultaneously, could then be attributed to the lack of coherence in the relevant cell-activations.

The "rhythm module" can also be improved: it is currently "hard-wired" in many places, and the rôle of learning is minimal. We should be careful, however, not to waste time modelling the learning of faculties which appear to be innate *in vivo*, i.e. faculties which have been "learned" at the evolutionary level. The validity of the rhythm module could also be assessed by direct comparison with Subjects' performance at beat-tracking tasks.

Apart from the developing the network models themselves, there is considerable scope for improvement in the computer simulations of their behaviour. In particular, the current simulations run quite slowly on the hardware currently available, to the extent that wider application and further development of the network might be adversely affected. The intrinsically "parallel" nature of many of the calculations performed during simulation, suggests that distribution of the computational load across many parallel processors might provide a solution to the problem of excessive run-time. In the long run, construction of, for instance, "silicon neurons" with the desired characteristics might be a viable alternative. I estimate that a hundredfold speed-up in run-time, would allow the network to perform in real-time, at least for the melody-learning tasks described here.

Speeding up the network simulations will permit their wider application, by allowing larger (and deeper) networks to be simulated. The resulting networks could then be applied not only to the further modelling of aspects of music perception, with much larger input sets etc., but also to the modelling of, for instance, aspects of language perception. Such simulations might reveal common mechanisms underlying the perception of both stimulus types. Investigation of the rhythmic aspects of language perception is of particular interest — Cutler [31] cites evidence that the rhythmic pattern of a language aids in "bootstrapping" the process of speech segmentation, in a manner similar to that suggested here for melody segmentation.

Finally, further work is needed to specify the ways in which a larger network might be built from many of the "modules" developed here. In particular, the interaction between parallel hierarchies, describing different aspects (pitches, rhythm etc.) of the same stimulus sequence, needs to be investigated.

Bibliography

[1] Abe and Hoshino. Schema driven properties in melody cognition: Experiments on final tone extrapolation by music experts. *Psychomusicology*, 9(2):161–172, 1990.

[2] D. Allen. Octave discriminability of musical and non-musical subjects. *Psychonomic Science*, 7:421–422, 1967.

[3] Paul E. Allen and Roger B. Dannenberg. Tracking musical beats in real time. In *Proceedings of the International Computer Music Conference, Glasgow*, pages 140–143. Computer Music Society, 1990.

[4] M.W. Andrews and W.J. Dowling. The development of perception of interleaved melodies and control of auditory attention. *Music Perception,*, 8(4):349–368, 1991.

[5] F. Attneave and R. R. Olson. Pitch as medium: A new approach to psychophysical scaling. *American Journal of Psychology*, 84:147–166, 1971.

[6] J. C. Bartlett and W. J. Dowling. Scale structure and similarity of melodies. *Music Perception*, 5:285–314, 1988.

[7] M. I. Bellgard and C. P. Tsang. Harmonizing music using a network of Boltzmann machines. In *Proceedings of the Fifth Annual Conference of Artificial Neural Networks and their Applications, Neuro-Nimes, Nimes, France*, pages 321–332, November 1992.

[8] J. J. Bharucha. Event hierarchies, tonal hierarchies, and assimilation: A reply to Deutsch and Dowling. *Journal of Experimental Psychology: General*, 113(3):412–425, 1984.

[9] J. J. Bharucha. Music cognition and perceptual facilitation: A connectionist framework. *Music Perception*, 5(1):1–30, 1987.

[10] J. J. Bharucha. Neural nets and perceptual learning of tonal expectancies. In *Proceedings of the First International Conference on Music Perception and Cognition, Kyoto*, pages 81–86, 1989.

[11] J. J. Bharucha and K. Stoeckig. Reaction time and musical expectancy: Priming of chords. *Journal of Experimental Psychology: Human Perception and Performance*, 12(4):403–410, 1986.

[12] J. J. Bharucha and K. Stoeckig. Priming of chords: Spreading activation or overlapping frequency spectra. *Perception and Psychophysics*, 41(6):519–524, 1987.

[13] Jamshed J. Bharucha. Neural net modeling of music. In *Proceedings of the First Workshop on Artificial Intelligence and Music*, St. Paul, Minnesota, 1988. American Association of Artificial Intelligence.

[14] Jamshed J. Bharucha. Pitch, harmony and neural nets: A psychological perspective. In P. M. Todd and D. Gareth Loy, editors, *Music and Connectionism*, pages 84–99. MIT Press, Cambridge, Massachusetts, 1991.

[15] Jamshed J. Bharucha and Peter M. Todd. Modeling the perception of tonal structure with neural nets. *Computer Music Journal*, 13(4), 1989.

[16] Marilyn Boltz and Mari Riess Jones. Does rule recursion make melodies easier to produce? If not, what does? *Cognitive Psychology*, 18:389–431, 1986.

[17] G. H. Bower and F. Springston. Pauses as recoding points in letter series. *Journal of Experimental Psychology*, 83(3):421–430, 1970.

[18] A. S. Bregman. *Auditory Scene Analysis*. MIT Press, Cambridge, Massachusetts, 1990.

[19] H. Bruhn. *Harmonielehre als Grammatik der Musik*. Psychologie Verlags Union, Munich, 1988.

[20] J. C. Carlsen. Some factors which influence melodic expectancy. *Psychomusicology*, 1(1):12–29, 1981.

[21] G. A. Carpenter and S. Grossberg. A neural theory of circadian rhythms: The gated pacemaker. *Biological Cybernetics*, 48:35–59, 1983.

[22] G. A. Carpenter and S. Grossberg. ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26:4919–4930, 1987.

[23] G. A. Carpenter and S. Grossberg. ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, 3:129–152, 1990.

[24] G. A. Carpenter, S. Grossberg, and J. H. Reynolds. ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4, 1991.

[25] G. A. Carpenter, S. Grossberg, and D. B. Rosen. ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks*, 4(4):493–504, 1991.

[26] M. A. Castellano, J. J. Bharucha, and C. L. Krumhansl. Tonal hierarchies in the music of north india. *Journal of Experimental Psychology: General*, 113(3):394–412, 1984.

[27] Geoffrey J. Chappell and John G. Taylor. The temporal Kohonen map. *Neural Networks*, 6(3):441–445, 1993.

[28] E. F. Clarke. Structure and expression in rhythmic performance. In P. Howell, I.Cross, and R. West, editors, *Musical Structure and Cognition*, chapter 9, pages 209–236. Academic Press, London, 1985.

[29] M. A. Cohen and S. Grossberg. Masking fields: A massively parallel neural architecture for learning, recognizing, and predicting multiple groupings of patterned data. *Applied Optics*, 26(10):1866–1891, 1987.

[30] F. Crick. The recent excitement about neural networks. *Nature*, 337:129–132, 1989.

[31] A. Cutler. Cross-linguistic differences in speech segmentation. *MRC News*, 56:8–9, September 1992.

[32] Roger B. Dannenberg and Bernard Mont-Reynaud. Following an improvisation in real time. In *Proceedings of the International Computer Music Conference*, pages 241–248. Computer Music Association, 1987.

[33] Peter Desain. A (de)composable theory of rhythm perception. *Music Perception*, 9(4):439–454, 1992.

[34] Peter Desain and Henkjan Honing. The quantization of musical time: A connectionist approach. *Computer Music Journal*, 13(3), 1989.

[35] D. Deutsch. Music recognition. *Psychological Review*, 76(3):300–307, 1969.

[36] D. Deutsch. Octave generalization and tune recognition. *Perception and Psychophysics*, 11(6):411–412, 1972.

[37] D. Deutsch. Octave generalization and the consolidation of melodic information. *Canadian Journal of Psychology*, 33(3):201–205, 1979.

[38] D. Deutsch. The processing of structured and unstructured tonal sequences. *Perception and Psychophysics*, 28(5):381–389, 1980.

[39] D. Deutsch. The processing of pitch combinations. In D. Deutsch, editor, *The Psychology of Music*, pages 271–316. Academic Press, New York, 1982.

[40] D. Deutsch. Two issues concerning tonal hierarchies: Comment on Castellano, Bharucha, and Krumhansl. *Journal of Experimental Psychology: General*, 113(3):413–416, 1984.

[41] D. Deutsch and J. Feroe. The internal representation of pitch sequences in tonal music. *Psychological Review*, 88(6):503–522, 1981.

[42] W. J. Dowling. Recognition of melodic transformations: Inversion, retrograde, and retrograde inversion. *Perception and Psychophysics*, 12(5):417–421, 1972.

[43] W. J. Dowling. The perception of interleaved melodies. *Cognitive Psychology*, 5:322–337, 1973.

[44] W. J. Dowling. Scale and contour: Two components of a theory of memory for melodies. *Psychological Review*, 85:341–354, 1978.

[45] W. J. Dowling. Context effects on melody recognition: Scale-step versus interval representations. *Music Perception*, 3(3):281–296, 1986.

[46] W. J. Dowling. Tonal strength and melody recognition after long and short delays. *Perception and Psychophysics*, 50(4):305–313, 1991.

[47] W. J. Dowling and J. C. Bartlett. The importance of interval information in long-term memory for melodies. *Psychomusicology*, 1(1):30–49, 1981.

[48] W. J. Dowling and D. S. Fujitani. Contour, interval and pitch recognition in memory for melodies. *The Journal of the Acoustical Society of America*, 49(2):524–531, 1971.

[49] W. J. Dowling, K. M. Lung, and S. Herrbold. Aiming attention in pitch and time in the perception of interleaved melodies. *Perception and Psychophysics*, 41(6):642–656, 1987.

[50] W.J. Dowling. Rhythmic groups and subjective chunks in memory for melodies. *Perception and Psychophysics*, 14(1):37–40, 1973.

[51] C. Drake, W.J. Dowling, and C. Palmer. Accent structures in the reproduction of simple tunes by children and adult pianists. *Music Perception*, 8(3):315–334, 1991.

[52] G. M. Edelman. *Neural Darwinism: The Theory of Neuronal Group Selection.* Basic Books, New York, 1987.

[53] J. Edworthy. Melodic contour and musical structure. In P. Howell, I. Cross, and R. West, editors, *Musical structure and Cognition*, pages 169–188. Academic Press, London, 1985.

[54] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[55] Peter Essens. Hierarchical organization of temporal patterns. *Perception and Psychophysics*, 40:69–73, 1986.

[56] W. K. Estes. An associative basis for coding and organization in memory. In A. W. Melton and E. Martin, editors, *Coding Processes in Human Memory.* John Wiley, New York, 1972.

[57] Scott E. Fahlman. The recurrent cascade-correlation architecture. Technical Report CMU-CS-91-100, School of Computer Science, Carnegie Mellon University, 1991.

[58] Allen Forte. Schenker's conception of musical structure. *Journal of Music Theory*, 3(1):1–30, 1959.

[59] P. Fraisse. Rhythm and tempo. In D. Deutsch, editor, *The Psychology of Music*. Academic Press, New York, 1982.

[60] C. Gerard and C. Auxiette. The role of melodic and verbal organization in the reproduction of rhythmic groups by children. *Music Perception*, 6(2):173–192, 1988.

[61] J-Ph. Gervais and M. Greenhough. A microcomputer based system for evolving musical structures. *Proceedings of the Institute of Acoustics (Reproduced Sound 3)*, pages 133–140, 1987.

[62] Robert O. Gjerdingen. Meter as a mode of attending: A network simulation of attentional rhythmicity in music. *Intégral*, 3:67–92, 1989.

[63] Robert O. Gjerdingen. Using connectionist models to explore complex musical patterns. *Computer Music Journal*, 13(3), 1989.

[64] Robert O. Gjerdingen. Categorization of musical patterns by self-organizing neuronlike networks. *Music Perception*, 8:339–370, 1990.

[65] Robert O. Gjerdingen. Learning syntactically significant temporal patterns of chords: A masking field embedded in an ART3 architecture. *Neural Networks*, 5(4):551–564, 1992.

[66] J. L. Goldstein. An optimum processor theory for the central formation of the pitch of complex tones. *Journal of the Acoustical Society of America*, 54:1496–1516, 1973.

[67] Niall Griffith. Representing the tonality of musical sequences using neural nets. In *Proceedings of the International Conference on Music and Cognition*, Jyvskala, Finland, August 1993.

[68] S. Grossberg. Behavioral contrast in short term memory: Serial binary memory models or parallel continuous memory models. *Journal of Mathematical Psychology*, 17:199–219, 1978.

[69] S. Grossberg. The adaptive self-organization of serial order in behaviour: Speech, language and motor control. In E. C. Schwab and H. C. Nusbaum, editors, *Pattern Recognition by Humans and Machines: Speech Perception*, volume 1, chapter 6, pages 187–293. Academic Press, 1986.

[70] S. Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11, 1987.

[71] S. Grossberg. Nonlinear neural networks: Principles, mechanisms and architectures. *Neural Networks*, 1:17–61, 1988.

[72] S. Grossberg and M. Kuperstein. *Neural Dynamics of Adaptive Sensory-Motor Control*, chapter 6. Elsevier Science Publishers, Amsterdam, North Holland, 1986.

[73] S. Grossberg and D. Somers. Synchronized oscillations during cooperative feature linking in a cortical model of visual perception. *Neural Networks*, 4(4):453–467, 1991.

[74] Stephen Handel and James S. Oshinsky. The meter of syncopated auditory polyrhythms. *Perception and Psychophysics*, 30(1):1–9, 1981.

[75] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its applications to conduction and excitation in nerve. *J Physiol. London*, 117, 1952.

[76] Paul Horowitz and Winfield Hill. *The Art of Electronics*. Cambridge University Press, Cambridge, 1980.

[77] George Houghton. The problem of serial order: A neural network model of sequence learning and recall. In R. Dale, C. Mellish, and M. Zock, editors, *Current Research in Natural Language Generation*, pages 287–319. Academic Press, London, 1990.

[78] D.H. Hubel and T.N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 24:119–128, 1961.

[79] D. Huron. Tonal consonance versus tonal fusion in polyphonic sonorities. *Music Perception*, 9(2):135–154, 1991.

[80] W. I. Idson and D. W. Massaro. A bidimensional model of pitch in the recognition of melodies. *Perception and Psychophysics*, 24(6):551–565, 1978.

[81] R. Jackendoff. Musical parsing and musical affect. *Music Perception*, 9(2):199–230, 1991.

[82] N. F. Johnson. The role of chunking and organization in the process of recall. In G. H. Bower, editor, *Psychology of Learning and Motivation*. Academic Press, New York, 1970.

[83] M. R. Jones. Time, our lost dimension: Toward a new theory of perception, attention and memory. *Psychological Review*, 83:323–355, 1976.

[84] M. R. Jones. Music as a stimulus for psychological motion: Part i. some determinants of expectancies. *Psychomusicology*, 1(2):34–51, 1981.

[85] M. R. Jones. Music as a stimulus for psychological motion: Part ii: An expectancy model. *Psychomusicology*, 2(1):1–13, 1982.

[86] Mari Riess Jones. Dynamic pattern structure in music. *Perception and Psychophysics*, 41(6):621–634, 1987.

[87] Mari Riess Jones and Marilyn Boltz. Dynamic attending and responses to time. *Psychological Review*, 96(3):459–491, 1989.

[88] M. I. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Hillsdale, N.J., 1986.

[89] H. J. Kallman and D. W. Massaro. Tone chroma is functional in melody recognition. *Perception and Psychophysics*, 26(1):32–36, 1979.

[90] M. Kalmar and G. Balasko. Musical mother tongue and creativity in preschool children's melody improvisations. *Bulletin for the Council for Research in Music Education*, 91:77–8, 1987.

[91] Bruce F. Katz. Musical pleasure through the unification of melodic sequences. In *Proceedings of the Cognitive Science Conference, Boulder*, 1993.

[92] Bruce F. Katz. Musical resolution and musical pleasure. In *Proceedings of AISB*. Society for Artificial Intelligence and Simulation of Behaviour, 1993.

[93] David Kleinfeld. Sequential state generation by model neural networks. *Proceedings of the National Academy of Sciences, USA*, 83:9469–9473, December 1986.

[94] A. Koestler. *The Sleepwalkers*. Hutchinson and Co. Ltd, London, 1950.

[95] Teuvo Kohonen, Pauli Laine, Kalev Tiits, and Kari Torkkola. A nonheuristic automatic composing method. In P. M. Todd and D. Gareth Loy, editors, *Music and Connectionism*, pages 229–242. MIT Press, Cambridge, Massachusetts, 1991.

[96] Bart Kosko. Bidirectional associative memories. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):49–60, 1988.

[97] Bart Kosko. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence.* Prentice Hall Inc., Englewood Cliffs, N.J., 1990.

[98] C. L. Krumhansl and E. J. Kessler. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review*, 89(4):334–368, 1982.

[99] C. L. Krumhansl and R. N. Shepard. Quantification of the hierarchy of tonal functions within a diatonic context. *Journal of Experimental Psychology: Human Perception and Performance*, 5:579–594, 1979.

[100] Bernice Laden and Douglas H. Keefe. The representation of pitch in a neural net model of chord classification. *Computer Music Journal*, 13(4), 1989.

[101] Kevin J. Lang, Alex H. Waibel, and Geoffrey E. Hinton. A time-delay neural network architecture for isolated word recognition. *Neural Networks*, 3(1):23–43, 1990.

[102] E. W. Large, C. Palmer, and J. Pollack. A connectionist model of intermediate representations for musical structure. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 412–417, Hillsdale, N.J., 1991. Erlbaum Press.

[103] C. S. Lee. The rhythmic interpretation of simple musical sequences: Towards a perceptual model. In P. Howell, I.Cross, and R. West, editors, *Musical Structure and Cognition*, chapter 3, pages 53–69. Academic Press, London, 1985.

[104] Marc Leman. The ontogenesis of tonal semantics: Results of a computer study. In P. M. Todd and D. Gareth Loy, editors, *Music and Connectionism*, pages 100–127. MIT Press, Cambridge, Massachusetts, 1991.

[105] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music.* MIT Pres, Cambridge, Massachusetts, 1983.

[106] J. P. Lewis. Creation by refinement and the problem of algorithmic music composition. In P. M. Todd and D. Gareth Loy, editors, *Music and Connectionism*, pages 243–255. MIT Press, Cambridge, Massachusetts, 1991.

[107] C. Linster. A neural network that learns to play in different styles. In *Proceedings of the International Computer Music Conference, Glasgow*, pages 311–313. Computer Music Society, 1990.

[108] H. C. Longuet-Higgins and C. S. Lee. The perception of musical rhythms. *Perception*, 11:115–128, 1982.

[109] L. Meyer. *Emotion and Meaning in Music.* University of Chicago Press, Chicago, 1956.

[110] L. Meyer. *Explaining Music.* University of California Press, Berkeley, C.A, 1973.

[111] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, 1956.

[112] Bernard Mont-Reynaud and Mark Goldstein. On finding rhythmic patterns in musical lines. In *Proceedings of the International Computer Music Conference*, pages 391–397. Computer Music Association, 1985.

[113] Michael C. Mozer. Connectionist music composition based on melodic, stylistic, and psychophysical constraints. In P. M. Todd and D. Gareth Loy, editors, *Music and Connectionism*, pages 195–212. MIT Press, Cambridge, Massachusetts, 1991.

[114] Michael C. Mozer. Neural net architectures for temporal sequence processing. In A. Weigend and N. Gershfield, editors, *Predicting the Future and Understanding the Past.* Addison-Wesley, Redwood City, CA, 1993.

[115] E. Narmour. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model.* University of Chicago Press, Chicago, 1990.

[116] E. Narmour. The top-down and bottom-up systems of musical implication: Building on Meyer's theory of emotional syntax. *Music Perception*, 9(1):1–26, 1991.

[117] A. L. Nigrin. *The Stable Learning of Temporal Patterns with an Adaptive Resonance Circuit.* PhD thesis, Duke University, 1990.

[118] A. L. Nigrin. Sonnet 1: A self-organizing masking field that solves the stability-plasticity dilemma. 1992.

[119] Y. Oura. Constructing a representation of a melody: Transforming melodic segments into reduced pitch patterns operated on by modifiers. *Music Perception*, 9(2):251–266, 1991.

[120] M. P. A. Page. The musical expectations of self-organizing neural networks. In *Proceedings of the International Computer Music Conference, Montreal*, pages 206–209, 1991.

[121] Caroline Palmer and Carol L. Krumhansl. Pitch and temporal contributions to musical phrase perception: Effects of harmony, performance timing, and familiarity. *Perception and Psychophysics*, 41(6):505–518, 1987.

[122] Caroline Palmer and Carol L. Krumhansl. Mental representations for music meter. *Journal of Experimental Psychology: Human Perception and Performance*, 16(4):728–741, 1990.

[123] W. Piston. *Harmony.* Norton, New York, 1978.

[124] Robert F. Port. Representation and recognition of temporal patterns. *Connection Science*, 2(1-2):151–176, 1990.

[125] D. J. Povel and H. Okkerman. Accents in equitone sequences. *Perception and Psychophysics*, 30:565–572, 1981.

[126] Dirk-Jan Povel. Internal representation of simple temporal patterns. *Journal of Experimental Psychology: Human Perception and Performance*, 7(1):3–18, 1981.

[127] Dirk-Jan Povel and Peter Essens. Perception of temporal patterns. *Music Perception*, 2(4):411–440, 1985.

[128] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, 1988.

[129] Michael Reiss and John G. Taylor. Storing temporal sequences. *Neural Networks*, 4:773–787, 1991.

[130] F. Restle. Theory of serial pattern learning: Structural trees. *Psychological Review*, 77(6):481–495, 1970.

[131] F. Restle. Serial patterns: The rle of phrasing. *Journal of Experimental Psychology*, 92(3):385–390, 1972.

[132] F. Restle and E. R. Brown. Serial pattern learning. *Journal of Experimental Psychology*, 83:120–125, 1970.

[133] David Rosenthal. A model of the process of listening to simple rhythms. *Music Perception*, 6(3):315–328, 1989.

[134] David Rosenthal. Emulation of human rhythm perception. *Computer Music Journal*, 16(1):64–76, 1992.

[135] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, Cambridge, Massachusetts, 1986.

[136] Hajime Sano and B. Keith Jenkins. A neural network model for pitch perception. *Computer Music Journal*, 13(3), 1989.

[137] Samir I. Sayegh. Fingering for string instruments with the optimum path paradigm. *Computer Music Journal*, 13(3), 1989.

[138] Don L. Scarborough, Ben O. Miller, and Jacqueline A. Jones. Connectionist models for tonal analysis. *Computer Music Journal*, 13(3), 1989.

[139] Jurgen Schmidhuber. Neural sequence chunkers. Technical Report FKI-148-91, Institut fur Informatik, Technische Universitat Munchen, 1991.

[140] M. A. Schmuckler. Expectation in music: Investigation of melodic and harmonic processes. *Music Perception*, 7(2):109–150, 1989.

[141] M. A. Schmuckler. The performance of global expectations. *Psychomusicology*, 9(2):122–147, 1990.

[142] R. N. Shepard. Geometrical approximations to the structure of musical pitch. *Psychological Review*, 89:305–333, 1982.

[143] H. A. Simon and R. K. Sumner. Pattern in music. In B. Kleinmuntz, editor, *Formal Representation of Human Judgement*. Wiley, New York, 1968.

[144] J. A. Sloboda and D. H. H. Parker. Immediate recall of melodies. In P. Howell, I. Cross, and R. West, editors, *Musical structure and Cognition*, pages 143–167. Academic Press, London, 1985.

[145] S. W. Smoliar. Review of the analysis and cognition of basic melodic structures: The implication-realization model by E. Narmour. *In Theory Only*, 12(1-2):43–56, 1991.

[146] N. Tan, R. Aiello, and T.G. Bever. Harmonic structure as a determinant of melodic organization. *Memory and Cognition*, 9(5):533–539, 1981.

[147] E. Terhardt. Pitch, consonance and harmony. *Journal of the Acoustical Society of America*, 55:1061–1069, 1974.

[148] M. T. Thomassen. Melodic accent: Experiments and a tentative model. *Journal of the Acoustical Society of America*, 71:1596–1605, 1982.

[149] P. M. Todd. A connectionist approach to algorithmic composition. *Computer Music Journal*, 13(4), 1989.

[150] Peter M. Todd and D. Gareth Loy, editors. *Music and Connectionism*. MIT Press, Cambridge, Massachusetts, 1991.

[151] C. von der Malsburg. Pattern recognition by labeled graph matching. *Neural Networks*, 1:141–148, 1988.

[152] D. Vorberg and R. Hambuch. On the temporal control of rhythmic performance. In J. Requin, editor, *Attention and Performance VII*. Erlbaum, Hillsdale, NJ, 1978.

[153] Alex H. Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(3):328–339, 1989.

[154] Deliang Wang and Michael A. Arbib. Complex temporal sequence learning. *Proceedings of the IEEE*, 78(9):1536–1543, 1990.

[155] R. M. Warren, D. A. Gardner, B. S. Brubaker, and J. A. Bashford. Melodic and nonmelodic sequences of tones: Effects of duration on perception. *Music Perception*, 8(3):277–290, 1991.

[156] B. W. White. Recognition of distorted melodies. *American Journal of Psychology*, 73:100–107, 1960.

[157] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:281–294, 1989.