10
Computational Evolutionary Musicology

EDUARDO R. MIRANDA AND PETER M. TODD

10.1. Introduction

The beginning of Chapter 2 offered a sensible definition of music as *temporally organized sound*. In the broader sense of this definition, one could arguably state that music is not uniquely human. A number of other animals also seem to have music of some sort. Complex vocalizations can be found in many birds (Marler and Slabbekoorn 2004), as well as in mammals such as whales (Payne and McVay 1971) and bats (Behr and von Helversen 2004). In a chapter suggestively entitled 'Zoomusicologie' in the book *Musique, Mythe, Nature ou Les Dauphins d'Arion*, Mâche (1991) presents an interesting discussion on the formal sophistication of various birdcalls. Recently Holy and Guo (2005) demonstrated that the ultrasonic vocalizations that male mice produce when they encounter female mice or their pheromones have the characteristics of song. What is intriguing is that primates who are close related to humans are not as 'musical' as those mammals that are far more distantly related to us. This intriguing fact suggests that music might have evolved independently among various types of animals, at various degrees of sophistication. In this context, it would be perfectly plausible to suggest the notion that robots might also be able to evolve music.

In order to build systems for the emergence of music one needs to establish the factors that may shape the course of musical evolution, such as physiological and cognitive factors, including models of interaction.

The physiological factors comprise the sensors and actuators of interacting individuals. These involve models of the hearing system, body, limbs and so on. It may also involve models of the sensory-motor cortex and associated neural mechanisms involved in sensory-motor tasks. A discussion on the expertise for building these models is beyond the scope of this chapter; it includes fields such as Biomechanics (Zinkovsky, Sholuha and Ivanov 1996) and Biophysics (Glaser 2001). Readers are invited to consult the literature in these fields.

As for the cognitive factors, the brain certainly uses different mental modules to process music and these modules have varying degrees of independence from each other. Parsons (2003) has conducted a number of brain-imaging experiments, which indicate that the neural system for processing music is widely distributed throughout our brain. This finding is confirmed by studies of patients with brain lesions (Peretz et al. 1994). Peretz and Coltheart (2003) proposed a functional architecture of the brain for music processing that captures the typical properties of such distributed modular organization. Basically, they have identified two main processing modules: One concerned with processing pitch and the other with rhythm. Both modules process incoming musical signals backed by a musical lexicon; a kind of memory bank of musical segments. Surely, this basic architecture can be refined into smaller and perhaps more specialized components, depending on the level of detail at which one wishes to study its functionality. This is likely to become increasingly complex as research in the emerging field of Cognitive Neuroscience of Music progresses. What is important, however, is that this modularity of the brain for music processing suggests a plausible methodology for building computational models of music processing. By focusing on relatively simpler cognitive tasks, one can adopt a bottom up approach to tackle the problem of modelling cognitive factors. In most cases it may be more effective to address the individual pieces of the jigsaw first, rather than the whole picture at once.

Following the development of fields such as Artificial Life (Levy 1993) and Memetics (Brodie 1996), we propose a computational modelling approach to Evolutionary Musicology: *Computational Evolutionary Musicology*. Evolutionary Musicology is the branch of Biomusicology (Wallin 1991) that studies the origins and evolution of music (Wallin, Merker and Brown 2000). For example, it studies the question of animal song and selection pressures underlying the evolution of music in humans and possibly in other species.

In gross terms, whereas Artificial Life attempts to understand evolutionary processes via the computer simulation of life forms Memetics studies the replication, spread and evolution of information patterns called memes. A fundamental notion of Memetics is that cultural evolution can be modelled through the same basic principles of variation and selection that underly biological evolution. This implies a shift from genes as units of biological information to a new type of units of cultural information, which are the *memes*. Computational Evolutionary Musicology therefore involves the use of computer modelling and simulations to study the circumstances and mechanisms whereby music systems might originate and evolve in artificially created worlds inhabited by communities of interacting autonomous agents (e.g. software agents or robots).

Scholars throughout the ages have attempted to trace the origins of music. The book *Music and the Origins of Language*, by Thomas (1995) presents a review of the theories purported by philosophers of the French Enlightenment. For example, in his *Essai sur l'origine des langues* (*Essay on the origins of language*) the philosopher Rousseau (1990) described the earliest spoken languages as being composed of vocal inflexions such as warnings, cries for help and hunting-related shouts. In the beginning, he proposed, gestures were preferred to communicate rational ideas, whereas vocal utterances were primarily used to express feelings and passions. As human society grew in complexity, these vocal utterances needed to become more precise and less passionate. And as language followed the path of logical argumentation, the intonation of these primordial utterances evolved into music. Music, according to Rousseau, thus evolved from the sounds of passionate speech.

More recently, *The Singing Neanderthals: The Origins of Music, Language, Mind and Body*, by Mithen (2005), discusses this subject from an evolutionary perspective supported by archaeological evidence. Also, *The Origins of Music*, edited by Wallin and colleagues (Wallin, Merker and Brown 2000), collates a series of chapters expressing a diversity of theories and viewpoints.

Computational Evolutionary Musicology is in many ways akin to the computational modelling approach of Evolutionary Linguistics to study of the origins and evolution of language (Cangelosi and Parisi 2001; Christiansen and Kirby 2003), but its goals are more pragmatic, in the sense that they can readily inform the development of new technologies for music making. For instance, a better understanding of basic mechanisms of musical origins and evolution is of great importance for musicians looking for hitherto unexplored ways to create new music works with computers. Broadly speaking, current techniques for implementing generative music systems can be classified as *abstract algorithmic* or *music knowledge-based.* Abstract algorithmic techniques are suitable for generating music from the behaviour of algorithms that were not necessarily designed for music in the first instance, but embody pattern generation features that are suitable for producing musical materials. Such algorithms include fractals (Milicevic 1996) and chaotic systems (Bidlack 1992) to cite but two examples. Music knowledge-based techniques generate music using algorithms derived from or inspired by well-established music theory. Most of these systems can learn compositional procedures from given examples, adopting either a symbolic approach (Steedman 1984; Cope 1996; Papadopoulos and Wiggins 1998) or a connectionist (neural networks) approach (Todd and Loy 1991; Mozer 1994), depending on the way they store information about music. Hybrid systems also exist (Burton and Vladimirova 1997).

Both classes of techniques have their merits and pitfalls. Abstract algorithmic techniques tend to produce rather complex music, most of which may sound too remote from what the majority of people, including expert listeners, would consider musical. This is possibly so because abstract algorithmic music tends to lack the cultural references that people normally rely upon when listening to music. Conversely, knowledge-based techniques tend to produce pastiches of existing musical pieces, which often are of little interest for composers aiming to create new music; that is, music that is not based on mimicking existing pieces or well-known musical styles. Computational Evolutionary Musicology brings the merits of both approaches closer to each other by offering the possibility of evolving new musical systems informed by the same principles that might have helped to shape existing musical styles. Inspired by Casti's (1997) use of the term 'would-be worlds', Artificial Life's goal of looking at 'life as it could be,' we refer to these emerging new musical systems as 'would-be music' or 'music as it could be'.

In this chapter, we explore some examples of Computational Evolutionary Musicology that employ a range of Artificial Life-inspired computational approaches. We begin by describing a model for studying the role of mating-selective pressure in the evolution of musical taste. Next, we introduce a mimetic model for studying the evolution of musical lexicons in a community of autonomous robots furnished with a vocal synthesizer, a hearing apparatus and a memory device. Finally, we present the application of neural networks to evolving simple generative sequencing rules in a community of rhythm players and imitators.

10.2. Mating Selective Pressure and Surprise

Todd and Werner (1999) proposed a model for studying the role of sexual selection, specifically the selective pressure that comes from the processes of choosing mates, in the evolution of music. The model employs mating selective pressure to foster the evolution of fit composers of courting tunes. The model co-evolves ‘male’ composers who play simple musical tunes along with ‘female’ critics who judge these tunes and decide with whom to mate in order to produce the next generation of composers and critics.

Todd and Werner’s model is largely inspired by Charles Darwin’s theory of sexual selection as outlined in the book *The Descent of Man and Selection in Relation to Sex*, where he argued that male birdsong had evolved via a mechanism of female choice (Darwin 1992). Miller (2000) attempted to take this notion even further by arguing that the evolution of human music was shaped by sexual selection to function as a courtship display.

Each male composer holds a tune of 32 musical pitches from a set of 24 different pitches spanning two octaves. The female critics encode a transition-table that rates the transitions from one note to another in a heard tune. The table is a 24-by-24 matrix, where each entry represents the female’s expectation of the probability of one pitch following another in a song. Given these expectations, a critic can decide how well she likes a particular tune. When she listens to a composer, she considers the transition from the previous pitch to the current pitch for each note of the tune, gives each transition a score based on her transition table and adds those scores to come up with her final evaluation of the tune. Each critic listens to the tunes of a certain number of composers who are randomly selected; all critics hear the same number of composers. After listening to all the composers in her courting-choir, the critic selects as her mate the composer who produces the tune to which she gives the highest score. In this selective process, all critics will have exactly one mate, but a composer may have a range of mates from none to many, depending on whether his tune is unpopular with everyone or if he has a song that is universally liked by the critics. Each critic has one child per generation created via crossover and mutation with her chosen mate. This child will have a mix of the musical traits and preferences encoded in its mother and father. The sex of the child is randomly determined and a third of the population is removed at random after a mating session in order not to reach a population overflow.

From the many different scoring methods proposed to judge the tunes, the one that seems to produce the most interesting results is the method whereby critics enjoy being surprised. Here the critic listens to each transition in the tune individually, computes how much she expected the transition and subtracts this value from the probability that she attached to the transition she most expected to hear. For example, if a critic has a value 0.8 stored in her transition table for the A–E transition, whenever she hears a note A in a tune, she would expect a note E to follow it 80% of the time. If she hears an A–C transition, then this transition will be taken as a surprise because it violates the A–E expectation. A score is calculated for all the transitions in the tune (e.g. the expected probability of the heard A–C transition, which might be 0.1, is subtracted from the expected A–E transition probability of 0.8 to yield a surprise rating of 0.7) and the final sum registers how much surprise the critic experienced; that is, how much she likes the tune. What is interesting here is that this does not result in the composers generating random tunes all over the place. It turns out that in order to get a high surprise score, a tune must first build up expectations, by making transitions to notes that have highly anticipated notes following them and then violate these expectations, by not using the highly anticipated note. Thus there is constant tension between doing what is expected and what is unexpected in each tune, but only highly surprising tunes are rewarded.

Overall, this model has shown that the selection of co-evolving male composers who generate surprising tunes and female critics who assess these tunes according to their preferences, can lead to the evolution of structured melodies and the maintenance and continual turnover of tune diversity over time.

In addition to mating selective pressure, this model embodies an important cognitive trait for survival: The ability to identify an unexpected element in a sequence of sound events. The preference for surprising tunes reflects this ability, which is very sophisticated in humans, even to the extent that our brain does not require our attention to perform this task. Neuroscientists have reported a component of the auditory event-relation potential (ERP), called mismatch negativity (MMN), which is elicited by a deviant stimulus in a repetitive auditory event, even in the absence of attention. ERP is a stereotyped electrophysiological response to a stimulus detected with the electroencephalogram (EEG). MMN is normally detected between 100 and 200 ms after the odd stimulus is heard. Näätänen and colleagues (2001) suggested that different sounds develop their representation in the neurophysiological substrate of the auditory memory and the MMN indicates an attention-independent change detection mechanism.

In Todd and Werner's model, the composers are initiated with random tunes and the critics with transition tables set with probabilities calculated from given folk-tune melodies. There is, however, a puzzling fundamental question that has not been addressed in this model: Where could the expectations of the female critics come from if they are not to be built in by hand? Would it be possible to evolve such expectations from scratch? A model that may provide support for addressing this question is introduced next.

10.3. Social Bonding and Imitation: Evolution of Intonation

Miranda (2002b) proposed a mimetic model where a small community of interactive robots programmed with appropriate motor, auditory and cognitive skills can evolve a shared lexicon of sonic intonation patterns from scratch, after a period of spontaneous creation, adjustment and memory reinforcement. In this case, expectation is defined as a sensory-motor mechanism whereby the robots evolve vectors of motor control parameters to produce imitations of heard intonation patterns. The robots thus expect to hear pitch sequences that correspond to their evolved motor vectors.

Intonation is generally defined as the melody of speech; it is characterized by the variations in the pitch of a speaker's voice. The rationale for attempting to model the evolution of intonation patterns comes from the fact that intonation is fundamental for the development of vocal communication. There have been a number of research reports giving evidence that babies are born with an acute sensitivity to intonation (Locke 1993; Nazzi et al. 1998). This ability probably evolved due to the need for enhanced mother-infant interactions. Baby talk or infant-directed-speech, sounds like music due its exaggerated intonation, which helps babies and very young children to develop their linguistic ability. Mothers use baby talk to influence the behaviour and elicit emotions in pre-linguistic infants. Ultimately, those mothers whose intonation abilities made them more able to provide infant care and those infants who were receptive to such care, have gained a reproductive advantage.

Following this idea, Miranda's robots are programmed with two fundamental instincts:

a) To imitate what they hear
b) To foster social bonding

Imitation is defined here as the task of hearing an intonation pattern and activating the motor system to reproduce it. Sociability is assessed in terms of the similarity of the robots' repertoires. In other words, in order to be sociable a robot must form a repertoire that is similar to the repertoire of its peers. The intonations thus create a social identity for the robots.

The importance of imitation for evolution has gained much attention after the discovery of mirror neurons in the frontal lobes of macaque monkeys. Mirror neurons are neurons which fire both when an animal performs an action and when the animal observes the same action performed by another animal, especially of the same species. Thus, the neurons mirror the behaviour of another animal, as though the observers were themselves performing the action. These neurons have subsequently been observed in some birds and in other primates including humans (Rizzolatti and Craighero 2004). The mirror system is sometimes considered to represent a primitive version of a simulation heuristic that might underlie a theory of mind (Gallese and Goldman 1998); the notion of theory of mind will re-appear in our discussion later. Interestingly, while mirror neurons are present in macaque monkeys, these monkeys have not been observed to imitate each other's behaviour. It is questionable whether mirror neurons evolved for learning by imitation. Instead, their function might rather be to allow an individual to understand what another individual is doing or to recognize the other individual's action.

The rationale for programming the robots with a drive for social bonding is supported by research by neurobiologists such as Freeman (1995), who brings his knowledge of brain chemistry to support the notion that music plays an important role in social bonding. According to Freeman, the brain releases oxytocin in the basal forebrain during group music making and dancing. The theory goes that by loosening synaptic links associated with prior knowledge, this hormone clears the path for the acquisition of new knowledge by sharing and imitating behaviour in a group.

Mithen (2005) also supports the notion that joint music making forges a group identity with high emotional content. 'Hominids would have frequently and meticulously examined the likely intentions, beliefs, desires and feelings of other members of a group before deciding whether to cooperate with them. But on other occasions simply trusting them would have been more effective, especially if quick decisions were necessary.' Those individuals who suppressed their own self-identity and instead forged a group identity by joint music making had better chances to thrive than those individuals who tended to act selfishly.

10.3.1. The Robots

The robots are equipped with a voice synthesizer, a hearing apparatus and a memory device. The model was initially implemented with software agents and the original interaction algorithms were largely inspired by the work of Luc Steels (1997) on evolutionary language games. Drouet subsequently helped to implement the robotic version described below, with refinements to the interaction algorithms (Fig. 10.1) (Miranda and Drouet 2006).

The voice synthesizer is essentially implemented as a physical model of the human vocal mechanism (Boersma 1993; Miranda 2002a). The robots need to compute three vectors of parameters in order to produce intonations: Lung pressure, the width of the glottis and the length and tension of the vocal chords, represented here as $lung\_pressure(n)$, $interarytenoid(n)$ and $cricothyroid(n)$, respectively. As for the hearing apparatus, it employs short-term autocorrelation-based analysis to extract the pitch contour of a vocal sound (Miranda 2001). The algorithm features a parameter that defines the sensitivity of the auditory perception of the robots. In essence, this parameter regulates the resolution of the hearing apparatus by controlling the precision of the short-term autocorrelation analysis.

Essentially, a robot's memory stores its repertoire of intonations, but it also stores other information such as probabilities, thresholds and reinforcement parameters. They have two distinct modules to store intonations in their memories: A motor map and a perceptual map. The motor map stores information in terms of three vectors of motor (vocal) parameters and the perceptual map stores information in terms of pitch contour.

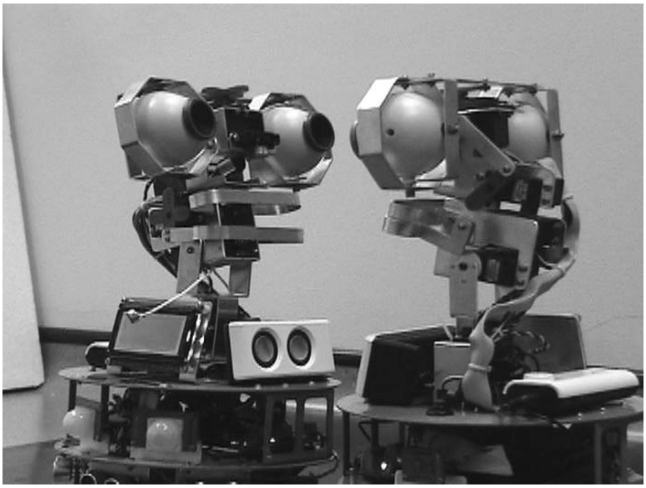


FIGURE 10.1. The robotic implementation uses DRK8000 robots, manufactured by Dr. Robot®.

An intonation is represented as a graph whose vertices stand for initial (or relative) pitch points and pitch movements and the edges represent a directional path. Whilst the first vertex must have one outbound edge, the last one must have only one incoming edge. All vertices in between must have one incoming and one outbound edge each. Vertices can be of two types, initial pitch points (referred to as *p-ini*) and pitch movements (referred to as *p-mov*) as follows (Fig. 10.2):

p-ini = {SM, SL, SH}
p-mov = {VLSU, LSU, MSU, SSU, RSB, SSD, MSD, LSD, VLSD}

where:

SM = start the intonation in the middle register
SL = start the intonation in the lower register
SH = start the intonation in the higher register

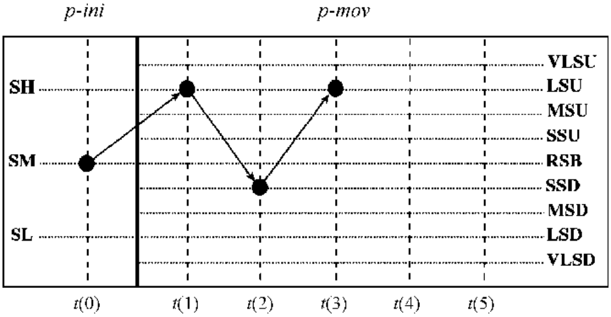


FIGURE 10.2. The representation of an intonation, where $t(n)$ indicates an ordered sequence of n pitches.

and

VLSU = very large step up
LSU = large step up
MSU = medium step up
SSU = small step up
RSB = remain at the same band
SSD = small step down
MSD = medium step down
LSD = large step down
VLSD = very large step down

An intonation will invariably start with a *p-ini*, followed by one or more *p-movs*. It is assumed that an intonation can start at three different voice registers: Low (SL), middle (SM) and high (SH). Then, from this initial point $\{t(n), n = 0\}$ the next pitch at $t(n + 1)$ might jump or step up or down and so forth.

It is important to note that labels or absolute pitch values are not relevant here because this scheme is intended to represent abstract melodic contours rather than a sequence of musical notes drawn from a specific tuning system. The tuning should emerge during the interactions.

10.3.2. The Algorithms

The main algorithms for the robotic interactions are given as follows:

Algorithm 1: Robot-player produces an intonation

```
Motor_control[$\alpha$] $\Leftarrow$ pick-any-motor-control in **Motor-Repertoire**(robot-player)
synthesize-sound(motor_control[$\alpha$])
```

Algorithm 2: Robot-imitator produces an imitation

```
Pitch_vector[$\beta$] $\Leftarrow$ perceive-intonation
Intonation[$\beta$] $\Leftarrow$ perceptual-representation(pitch_vector[$\beta$])
Intonation[$\Delta$] $\Leftarrow$ search-similar(intonation[$\beta$]) in **Perceptual-Repertoire** (robot-imitator)
Motor_control[$\Delta$] $\Leftarrow$ retrieve_motor_control(motor-control[intonation[$\Delta$])
synthesize-sound(motor_control[$\Delta$])
```

Algorithm 3: Robot-player hears the imitation and gives a feedback

```
Pitch_vector[$\psi$] $\Leftarrow$ perceive-imitation
Imitation[$\psi$] $\Leftarrow$ perceptual-representation(picth_vector[$\psi$])
Intonation[$\phi$] $\Leftarrow$ search-similar(imitation[$\psi$]) in **Perceptual-Repertoire** (robot-imitator)
Intonation[$\alpha$] = perceptual-representation(motor_control[$\alpha$])
IF intonation[$\alpha$] = intonation[$\phi$]
    THEN { feeback $\Leftarrow$ *positive*
            reinforce(motor_control[$\alpha$]) in **Motor-Repertoire**(robot-player)
            reinforce(intonation[$\alpha$]) in **Perceptual-Repertoire**(robot-player)}
    ELSE { feeback $\Leftarrow$ *negative* }
output-signal(feedback)
```

Algorithm 4: Robot-imitator reacts to robot-player's feedback

```
18. IF feedback = positive
19.     THEN { approximate(intonation[$\Delta$] $\rightarrow$ intonation[$\beta$])
                    in Perceptual-Repertoire(robot-imitator)
20.         reconfigure_motor_control(intonation[$\Delta$])
                    in Motor-Repertoire(robot-imitator)
21.         reinforce intonation[$\Delta$] in Perceptual-Repertoire(robot-imitator)
22.         reinforce motor_control($\Delta$) in Motor-Repertoire(robot-imitator) }
23.     ELSE IF feedback = negative
24.         THEN IF success-history(intonation[$\Delta$]) > success-threshold
25.             THEN{ motor_control[$\lambda$]$\Leftarrow$produce-new-motor-control
26.                 Intonation[$\lambda$]$\Leftarrow$ perceptual-representation
                        (motor_control[$\lambda$])
27.                 save-new(intonation[$\lambda$])
                        to Motor-Repertoire(robot-imitator)
28.                 save-new(motor_control[$\lambda$])
                        to Perceptual-Repertoire(robot-imitator) }
29.             ELSE { distantiate(intonation[$\Delta$] $\leftrightarrow$ intonation[$\beta$])
                        in Perceptual-Repertoire(robot-imitator)
30.                 reconfigure_motor_control(intonation[$\Delta$])
                        in Motor-Repertoire(robot-imitator) }
```

Algorithm 5: End of interaction updates

```
31. interaction-updates(robot-player)
32. interaction-updates(robot-imitator)
```

Figs. 10.3, 10.4 and 10.5 give a glimpse at the functioning of these algorithms. For didactic purposes, these are reduced two-dimensional representations of the motor and perceptual repertoires; the co-ordinates do not fully correspond to the actual motor and perceptual representations. The numbers in the figures indicate actions corresponding to the line numbers of the algorithms.

All robots have identical synthesis and listening apparatuses. At each round, each of the robots in a pair plays one of two different roles: The *robot-player* and the *robot-imitator*. The robot-player starts the interaction by producing an intonation α, randomly chosen from its repertoire. The robot-imitator then analyses the intonation α, searches for a similar intonation Δ in its repertoire and produces it. The robot-player in turn hears and analyses the intonation Δ and checks if its perceptual repertoire holds no other intonation ϕ that is more perceptibly close to Δ than α is. If it finds another intonation ϕ that is closer to Δ than α is, then the imitation is unsatisfactory, otherwise it is satisfactory. Fig. 10.3 shows an example where the robot-player and the robot-imitator hold in their memories two intonations each. The robot-player picks the intonation α from its motor-repertoire and produces it (1). The robot-imitator hears the intonation α and builds a perceptual representation β of it (4). Then it picks from its own perceptual repertoire the intonation Δ that is most perceptually similar to the heard intonation β (5) and produces it as an imitation (6). Next, the robot-player hears the imitation Δ and builds a perceptual representation ψ of it (9). Then it picks from its own perceptual

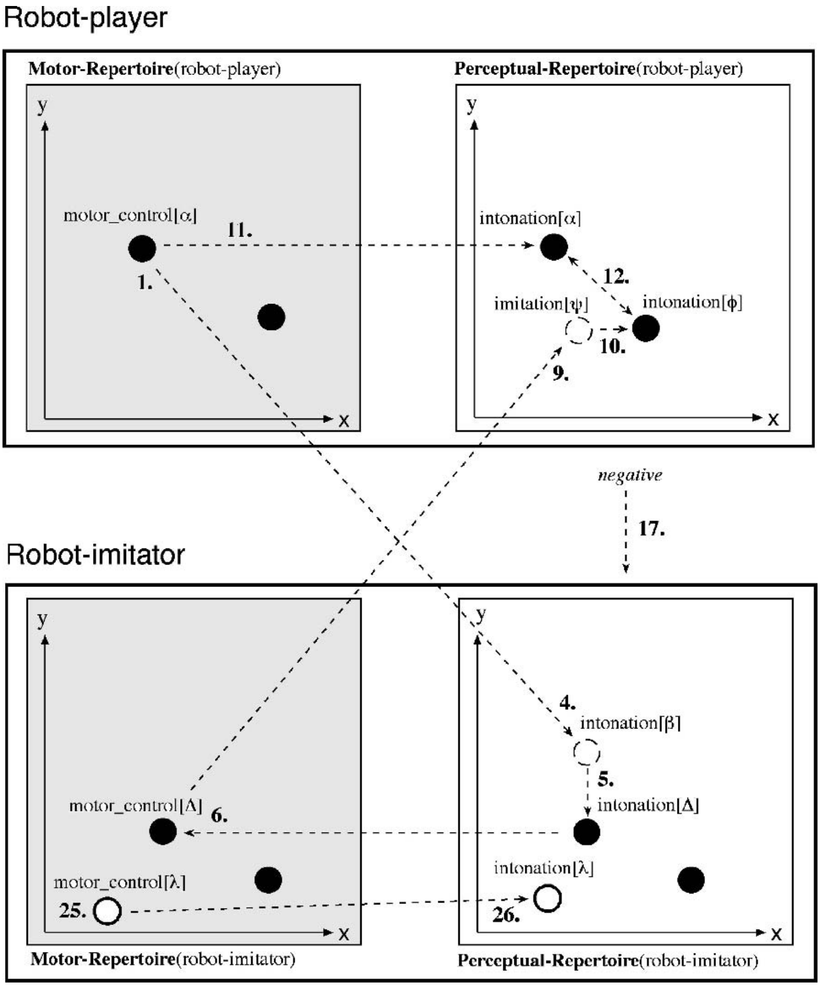


FIGURE 10.3. Example of an unsuccessful imitation.

repertoire the intonation ϕ that is most perceptually similar to the imitation ψ (10). The robot-player babbles the original intonation α to itself (11) and it concludes that α and ϕ are different (12). In this case the robot-player sends a negative feedback to the robot-imitator (17), indicating that the imitation is unsatisfactory.

When an imitation is unsatisfactory the robot-imitator has to choose between two potential courses of action. If it finds out that Δ is a weak intonation in its memory (because it has not received enough reinforcement in the past) then it will move it away from α slightly, as a measure to avoid repeating this mistake again. But if Δ is a strong intonation (due to a good past success rate), then the robot will leave Δ untouched (because it has been successfully used in previous imitations and a

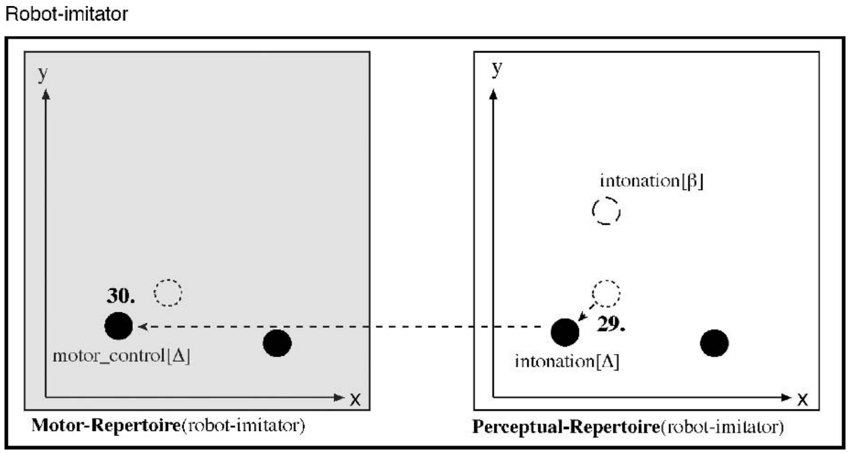

Robot-imitator
y
30.
motor_control[Δ]
x
Motor-Repertoire(robot-imitator)
y
intonation[β]
29.
intonation[Λ]
x
Perceptual-Repertoire(robot-imitator)

FIGURE 10.4. An example where the unsuccessful imitation involved an intonation that has a poor past success rate.

few other robots in the community also probably consider this intonation as being strong) and will create a new intonation λ similar to Δ to include it in its repertoire; that is, the robot produces a number of random imitations and then it picks the one that is perceptually most similar to Δ. Let us assume that the intonation Δ in Fig. 10.3 has a good past success rate. In this case, the robot-imitator leaves it untouched and creates a new intonation λ to include in its repertoire (25, 26).

Fig. 10.4 shows what would have happened if the intonation Δ did not have a good past success rate: In this case the robot-imitator would have moved Δ away from β slightly (29 and 30). Finally, Fig. 10.5 shows what would have happened

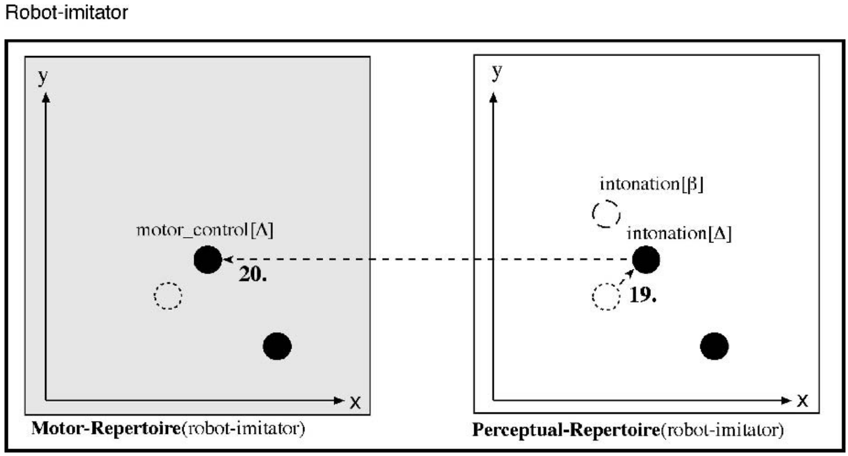

Robot-imitator
y
motor_control[Λ]
20.
x
Motor-Repertoire(robot-imitator)
y
intonation[β]
intonation[Δ]
19.
x
Perceptual-Repertoire(robot-imitator)

FIGURE 10.5. An example of a successful imitation.

if the robot-player had concluded that α and ϕ were the same, meaning that the imitation was successful. In this case, the robot-imitator would have reinforced the existence of the intonation Δ in its memory and would have moved it slightly towards the representation of the heard intonation β.

Before terminating the round, both robots perform final updates. Firstly they scan their repertoire and merge those intonations that are considered to be perceptibly close to each other; the merge function removes two intonations and creates a new one by averaging their values. Also, at the end of each round, both robots have a certain probability P_b of undertaking a spring-cleaning to get rid of weak intonations; those intonations that have not been sufficiently reinforced are forgotten. Finally, at the end of each round, the robot-imitator has a certain probability P_a of adding a new randomly created intonation to its repertoire; we refer to this coefficient as the 'creativity coefficient'.

10.3.3. A Typical Simulation Example

The graph in Fig. 10.6 shows a typical example of the evolution of the average repertoire of a group of five interacting robots, with snapshots taken after every 100 interactions over a total of 5000 interactions. The robots evolved repertoires averaging 12 intonations each. (Note that some may have developed more or less than 12 intonations.) After a drastic increase of the repertoire at about 800 interactions, the robots settled to an average of seven intonations each until about 2200 interactions, when another slight increase took place. Then they settled to an

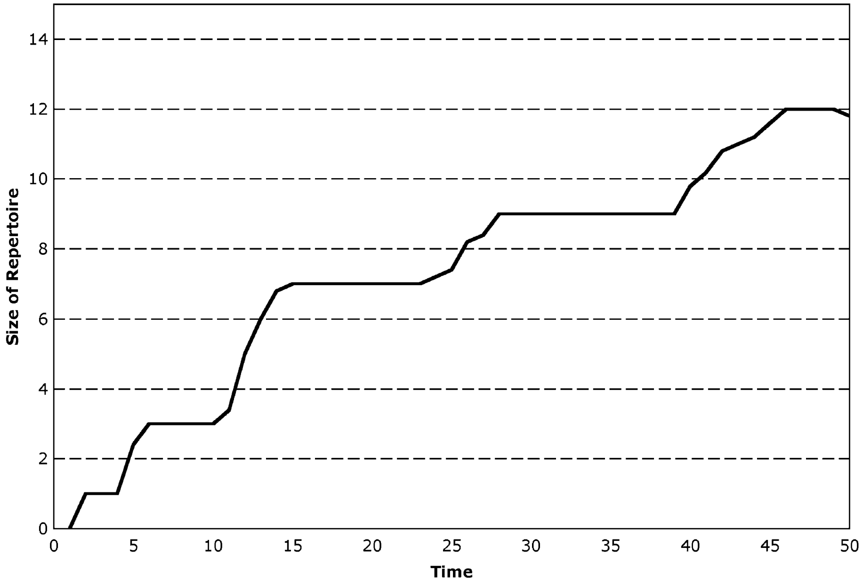


FIGURE 10.6. The evolution of the average size of the repertoire of intonations of the whole group of robots. In this case the group developed an average repertoire of 12 intonations.

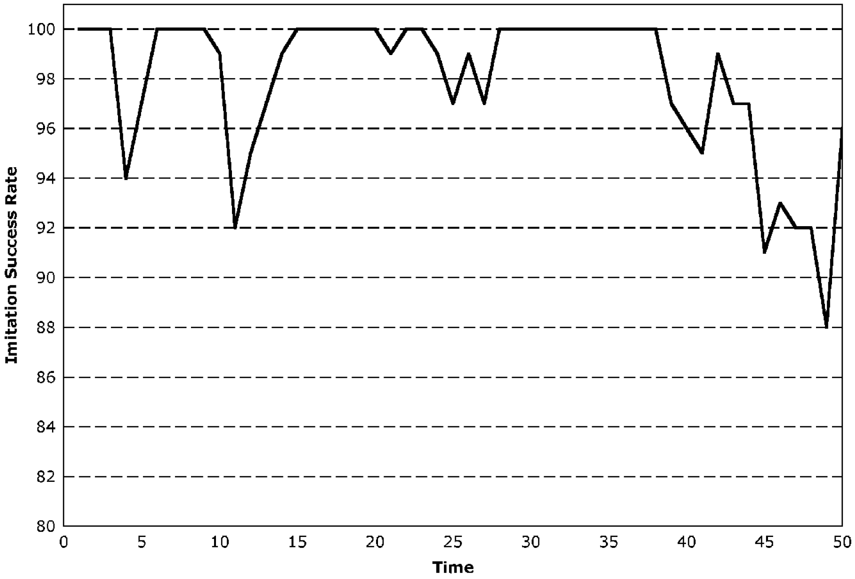


FIGURE 10.7. The imitation success rate over time.

average of nine intonations until about 3800 interactions. From 3800 interactions onwards the robots steadily increased their repertoires. The pressure to increase the repertoire is mostly due to the probability P_a of creating a new random intonation, combined with the rate of new inclusions due to unsatisfactory imitations. The size of the repertoire tends to stabilize with time because the more the robots use strongly settled intonations, the more these intonations are reinforced in their repertoires and therefore the more difficult for new intonations to settle in.

The graph in Fig. 10.7 plots the imitation success rate of the community, measured at every 100 interactions. Note the decrease of imitation success rate during those phases when the robots were increasing the size of their repertoires. Although the repertoire size tends to increase with time, the success rate tends to stay consistently high. However, this is highly dependent upon the number of robots in the group. The higher the number of robots, the deeper the fall of the success rate and the longer it takes to re-gain the 100% success rate stability.

Fig. 10.8(a) portrays the perceptual memory of a robot randomly selected from the group after 5000 interactions. In this case the length of the intonations varied from three to six pitches. (The minimum and maximum length of the intonation to be evolved is fixed beforehand.) This particular robot evolved 11 intonations; one below the average. Fig. 10.8(b) shows only those intonations that are three pitches long.

An interesting feature of this model is that the lexicon of intonations emerged from the interactions of the robots. The actions of each robot are based solely upon their own evolving expectations. Also, the robots do not necessarily have to evolve the same motor representations for what is considered to be perceptibly identical.

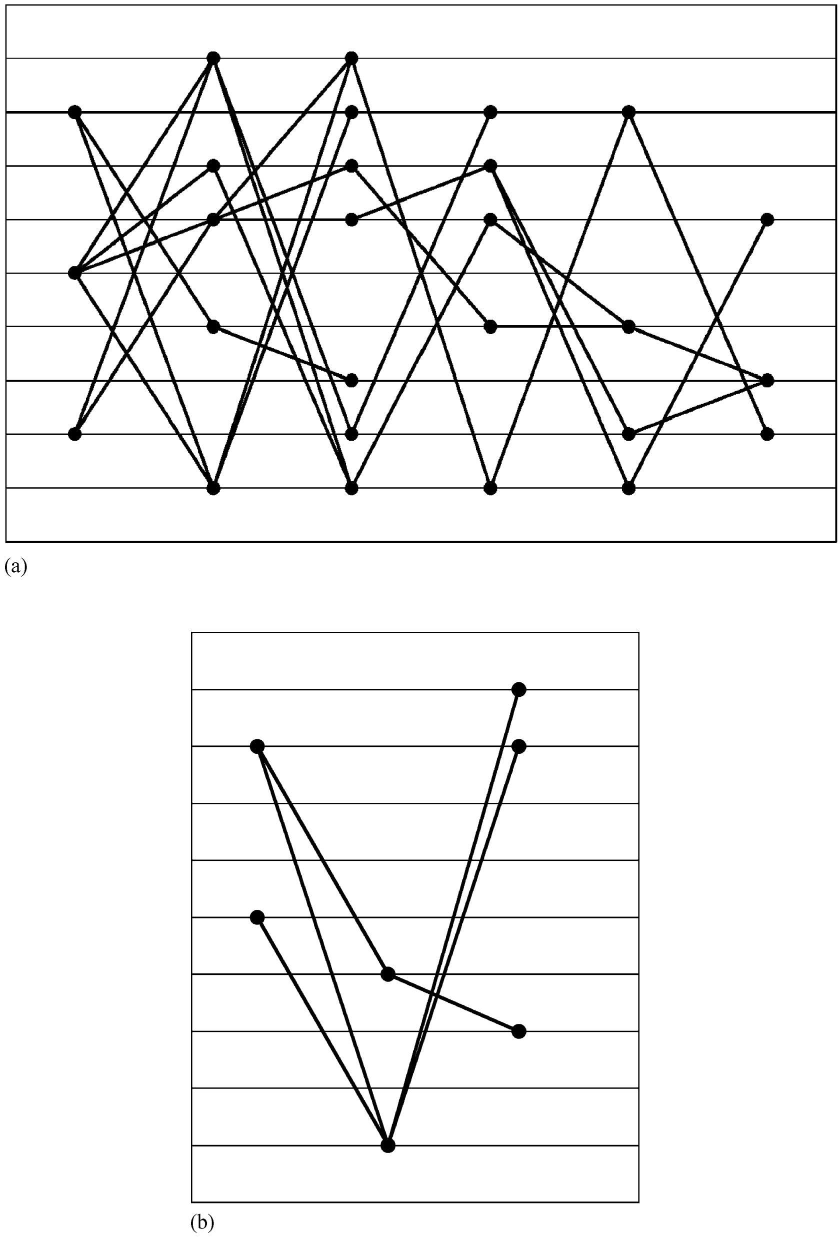
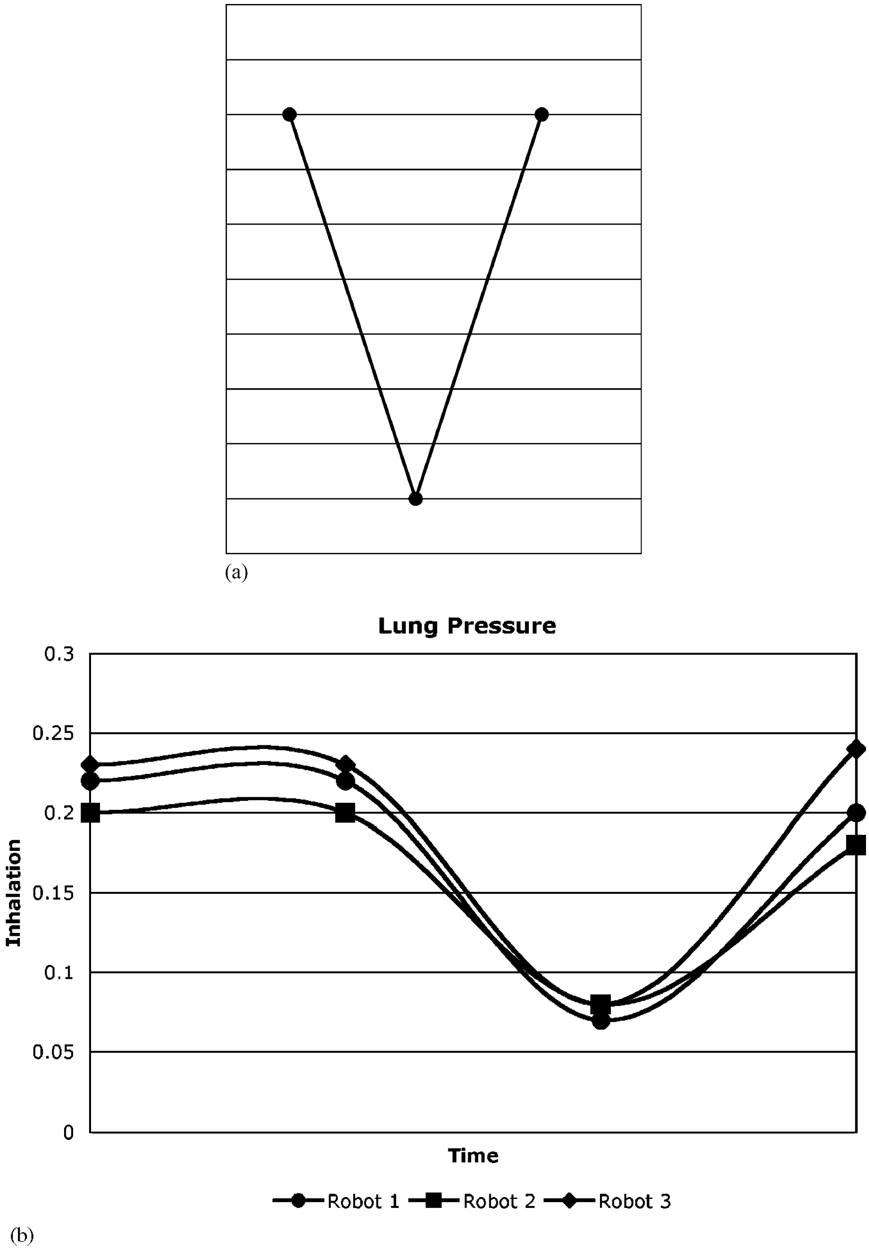


FIGURE 10.8. (a) The perceptual memory of one robot. (b) Only those intonations that are three pitches long. For the sake of clarity, the background metrics and labels of the graphs are not shown.



(a)

(b)

FIGURE 10.9. (a) One of the perceptual patterns from Fig. 10.8(b) and its corresponding motor control vectors developed by three different robots, (b) the *lung_pressure* vector, (c) the *cricothyroid* vector and (d) the *interarytenoid* vector.

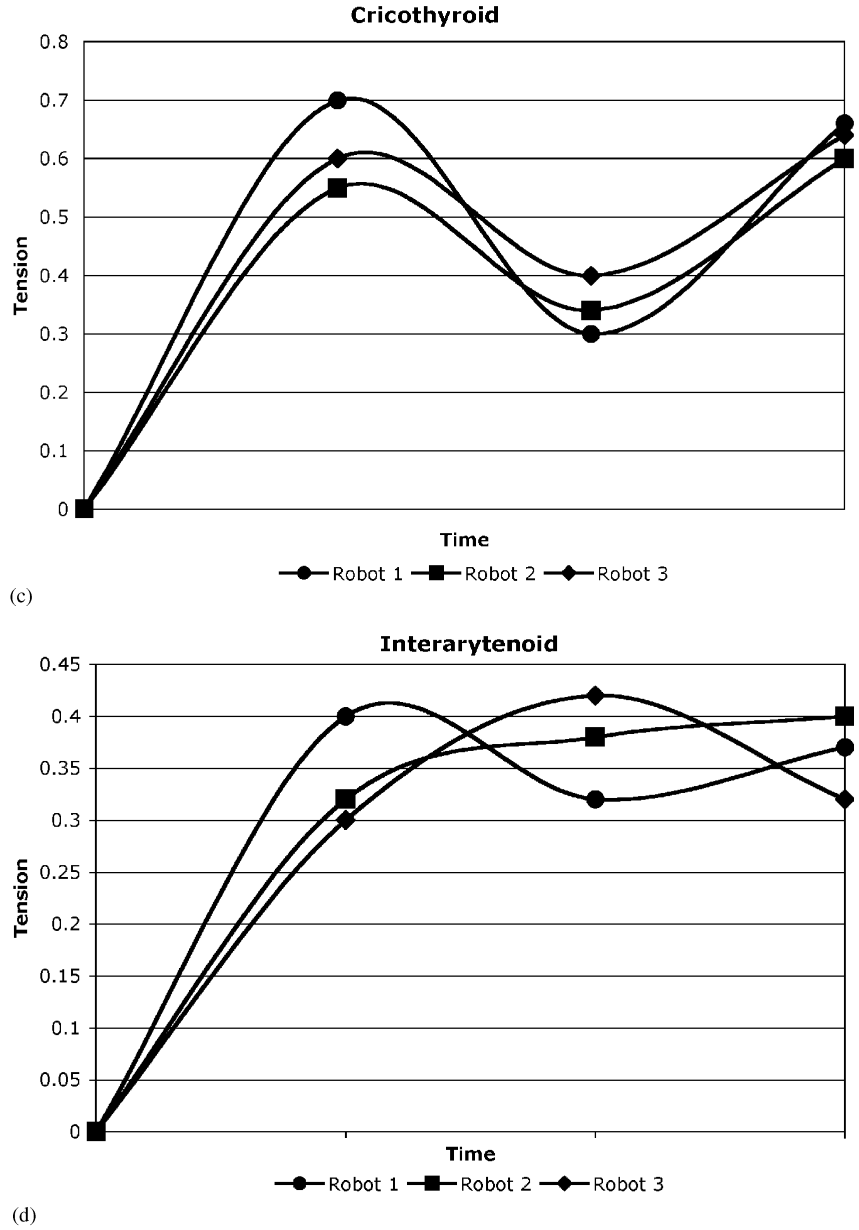


FIGURE 10.9. (*Continued*)

Fig. 10.9 shows the motor functions evolved by three different robots to represent what is essentially the same intonation.

The imitation of an intonation pattern requires the activation of the right motor parameters in order to reproduce it. The robot-imitators assume that they always can recognize everything they hear because in order to produce an imitation a robot will use the motor vectors that best match its perception of the sound in question. It is the robot-player who will assess the imitation based on its own expectations. Expectation here is a social convention but it is grounded on the nature of their sensory-motor apparatus. This mechanism provides a robot with the rudiments of the ability to appreciate the knowledge of another robot from its own perspective. Intuitively, this ability might relate to what is referred to as theory of mind. The notion of theory of mind is central to social life: The ability to understand that others have beliefs, desires and intentions that are different from one's own. The theory of mind probably emerged from the challenge of surviving in a complex social environment, in which decisions about cooperation were of vital importance. It is possible that possessing a theory of mind gave early hominids an evolutionary advantage over its ancestors, which was the ability to predict the behaviour of others and hence attain greater cooperation.

Both models discussed in this chapter so far deal with short intonations. But how about dealing with longer pitch sequences or proper musical compositions? Although the symbolic sensory-motor-like memory mechanism proposed for storing intonations served well the objectives of the model presented above, we must admit that it is not efficient for storing longer pitch sequences, let alone fully fledged pieces of music. In order to increase the complexity of the model, it is necessary to improve the memory mechanism, which would probably be more efficient by storing information about generating the sequences rather than the sequences themselves. Martins and Miranda (2006) are currently developing a connectionist approach to address this problem. Connectionism is an approach to modelling systems resembling biological neural networks whereby neurons are represented by nodes and connections between neurons are represented by links. The definition of the nature of the nodes and links determines the ability of the neural network to execute certain operations in a way that reproduces observed behaviours of the simulated biological system (Salu 2001).

10.4. Toward a Connectionist Memory for Evolving Sequencing Rules

João Martins developed a tentative connectionist memory for the aforementioned interacting robots consisting of two neural-network modules: A perceptual module and a categorization module. The former implements a Sardnet (for self-organizing activation, retention and decay network) neural network and the latter uses a feed-forward neural network.

The input for the perceptual module is a sequence of sounds, which produces a pattern of activations on the Sardnet, representing the types of sounds and their position in the sequence. The pattern of activations then becomes the input for the categorization module, which, as its name suggests, categorizes this information.

Each sound in the sequence presented to the Sardnet is represented as a vector of three variables. The first variable indicates its timbre, the second its loudness and the third the inter-onset interval (IOI) in milliseconds; pitch is not taken into account here. An inter-onset interval is the time between the beginnings or attack-points of successive sounds, not including their durations. For example, two 16th notes separated by a dotted eighth rest would have the same inter-onset interval as between a quarter note and a 16th note. As we are not so concerned with a detailed representation of timbre at this stage of the research, the value of the first variable is merely a label identifying the percussion instrument that played the sound; e.g. 1 = snare drum, 2 = bass drum, 3 = tom-tom, etc.

10.4.1. The Perceptual Module

The Sardnet is a self-organizing neural network for the classification of sequences (James and Miikkulainen 1995). The Sardnet is an extension of the self-organizing map (Som), which is a neural network used for unsupervised learning developed by Kohonen (1997). Fig. 10.10 shows a Som with 16 output nodes and one input vector $\mathbf{V}_t$, where t is the index of the sound event in the sequence. The dimension of $\mathbf{V}_t$ determines the dimension of the weights vector $\mathbf{W}_{jk}$ for each node. The Euclidean distance measure d_2 determines the distance from the input vector $\mathbf{V}_t$ to the weight vector $\mathbf{W}_{jk}$:

$$d_2(\mathbf{V}_t, \mathbf{W}_{jk}) = \sqrt{\sum_{i=1}^{n} \left| v_{t,i} - w_{jk,i} \right|^2}$$

where $\mathbf{V}_t$ is the input vector, with index t in the sequence, $\mathbf{W}_{jk}$ is the weight vector of the corresponding node jk and n is the dimension of both $\mathbf{V}_t$ and $\mathbf{W}_{jk}$.

The Som (Fig. 10.10) is also referred to as a competitive network or 'winner-takes-all net', because only the node whose weight vector is the closest to the input vector wins the activation. The weight vector of the winning node is subsequently updated in order to render its values even closer to the values of the input vector. The neighbouring nodes of the winning node are also similarly updated to a lesser degree according to a neighbourhood function that organizes representations of similar stimuli on the network topographically.

The Sardnet carries forward most of the essential features of the Som, but adds two important features, which enables it to deal with sequences of events:

a) The neuron that wins the competition for being closest to the input at one point in time is removed from subsequent competitions
b) All previous neuron activations decay at each time step

The dynamics of the Sardnet is illustrated in Fig. 10.11. Here a stream of events t at the input activated three nodes sequentially: $\mathbf{W}_{4,2}$, $\mathbf{W}_{2,3}$ and $\mathbf{W}_{1,2}$, respectively. The training algorithm for the Sardnet is as follows:

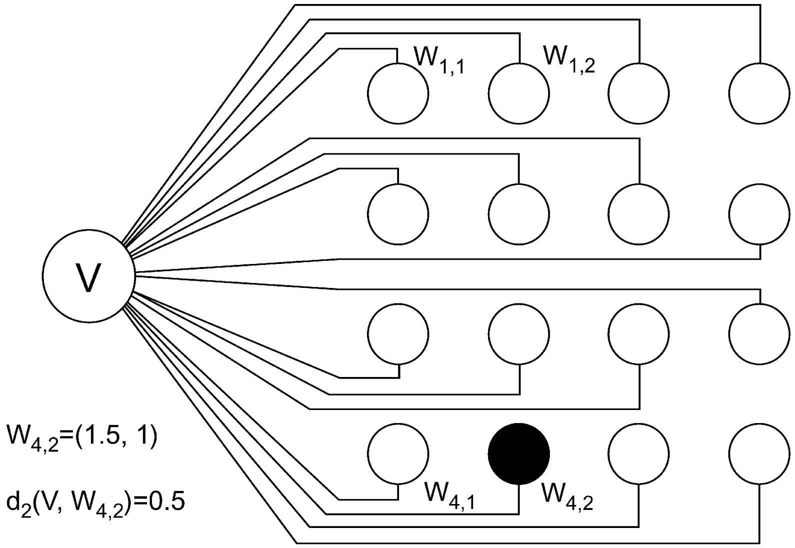


FIGURE 10.10. Kohonen's self-organizing map (Som). $\mathbf{V}_t$ is the input vector and the $\mathbf{W}_{j,k}$ are the weight vectors, which define the distance d_2 between the input vector and the various nodes of the network. In this example, the winner is node $\mathbf{W}_{4,2}$ with $d_2 = 0.5$.

```
INITIALIZATION:
    Reset the network
MAIN LOOP:
    While not end of sequence do:
    1. Find inactive neuron that best matches the input
    2. Assign activation = 1.0 to the found unit
    3. Adjust weight vectors of the neurons in the neighbourhood
    4. Exclude the winning neuron from subsequent competitions
    5. Decrement the activation values for all other active neurons
RESULT:
    Activated nodes ordered by activation values.
```

As with the Som network, the Sardnet uses the same distance $d_2(\mathbf{V}_t,\mathbf{W}_{jk})$ to estimate which node's weight vector best matches the input vector. In step 3 of the main loop of the training algorithm shown above, the weights of the winning neuron and of the neighbourhood neurons are changed according to the following adaptation rule:

$$\Delta w_{jk} = \alpha(w_{jk,i} - v_i)$$

where *j*and *k* are the spatial coordinates of the network, *i* is the index of the

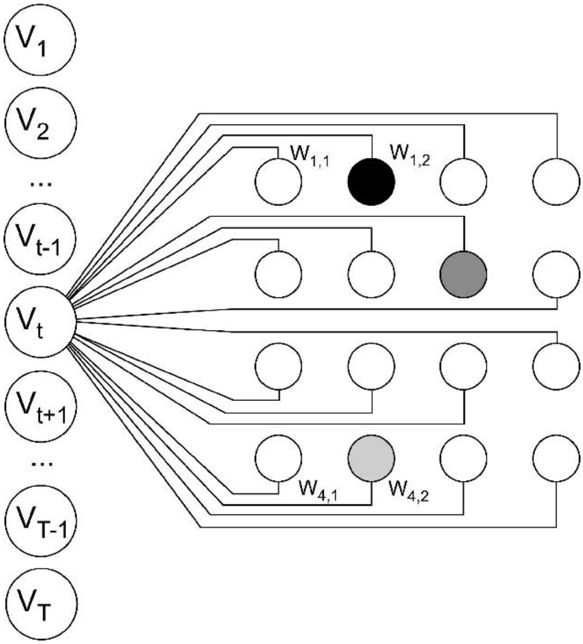


FIGURE 10.11. The Sardnet, with variable activation of three nodes sequentially, indicated by different colours: from black (node $\mathbf{W}_{1,2}$, the last in the sequence) to light grey (node $\mathbf{W}_{4,2}$, the first in the sequence).

individual components of the vectors and α denotes the learning rate of the Sardnet.

All active neurons are decayed proportionally to the decay parameter d as follows (step 5 of the main loop):

$$\eta_{jk}(t+1) = d\eta_{jk}(t), \qquad 0 < d < 1$$

where η is the value for the activation of the network in the element (j, k).

10.4.2. Categorization Module

The categorization module is a feedforward neural network (also called Multi-layer Perceptron) for learning patterns of activity with layers of nodes interconnected in a feed-forward way. Each input node is fully connected to the middle layer (referred to as the hidden layer) of nodes and each node of the hidden layer is subsequently connected to every output node (Fig. 10.12). The outputs of the network are explicit functions of activations in the hidden layer, which are themselves functions of the input nodes.

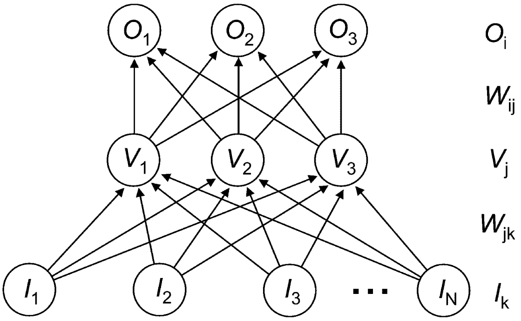


FIGURE 10.12. A generic feedforward neural network. I_n are the input nodes, V_n are the hidden nodes and O_n the output nodes. W_{jk} corresponds to the strengths (or weights) of the connections between the nodes j and k.

The network uses the backpropagation algorithm to adjust its weights in order to best match the values of the input nodes to a desired set of values at the output nodes.

The number of inputs to the categorization network must be the same as the number of units in the Sardnet, because each Sardnet unit becomes an input to the feedforward network. The number of output neurons is arbitrarily set to three because it facilitates the visualization of the resulting categorization in a tri-dimensional plotting.

10.4.3. Assessing the Behaviour of the Networks

As an example, let us consider an agent with a Sardnet of 50 nodes (10×5) with a learning rate $\propto = 0.1$. The network is initialized with random values for weight vectors in the range of -1 to 1. Assume that five rhythmic sequences played on one or two percussion instruments each (Fig. 10.13) are fed into the network a number of times.

After a few iterations, an organization pattern begins to emerge. The graphs in Fig. 10.14 show the evolution of the input weights corresponding to the inter-onset intervals (IOI) (the third component of the input vectors). Fig. 10.14(a) shows the initial value of the weights, as explained above and Fig. 10.14(b) shows the pattern of IOI weight values that emerged after 20 iterations. Then graph 14(c) shows the values of IOI weights that emerged after 80 iterations and 14(d) shows the difference between the sums of the weights on consecutive iterations.

Now, let us consider that the agent has a categorization network with 50 input units, whose values are given by the activations patterns of the Sardnet. These input nodes are fully connected to three nodes forming the hidden layer, which in turn are fully connected to three output nodes. This allows for straightforward visualization of the categorization of the rhythms in a tri-dimensional space. The first three activation layers of 50 Sardnet nodes corresponding to three rhythms were used to train the feedforward network to match them with three different

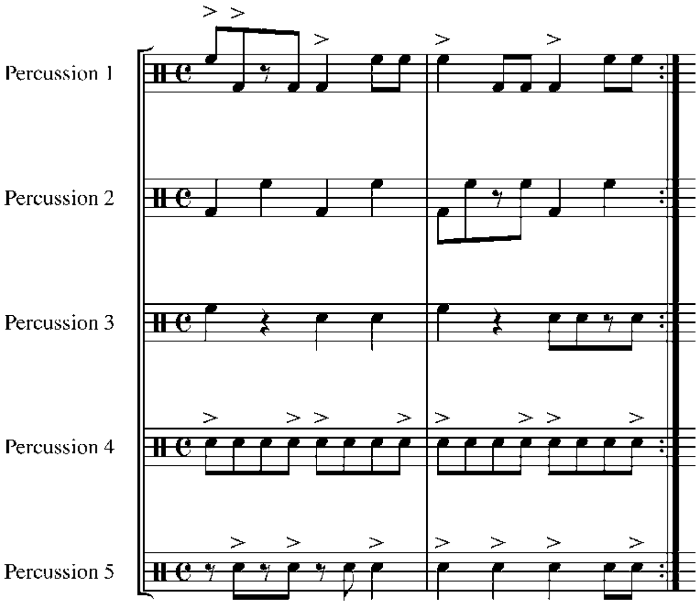


FIGURE 10.13. The five training rhythmic sequences.

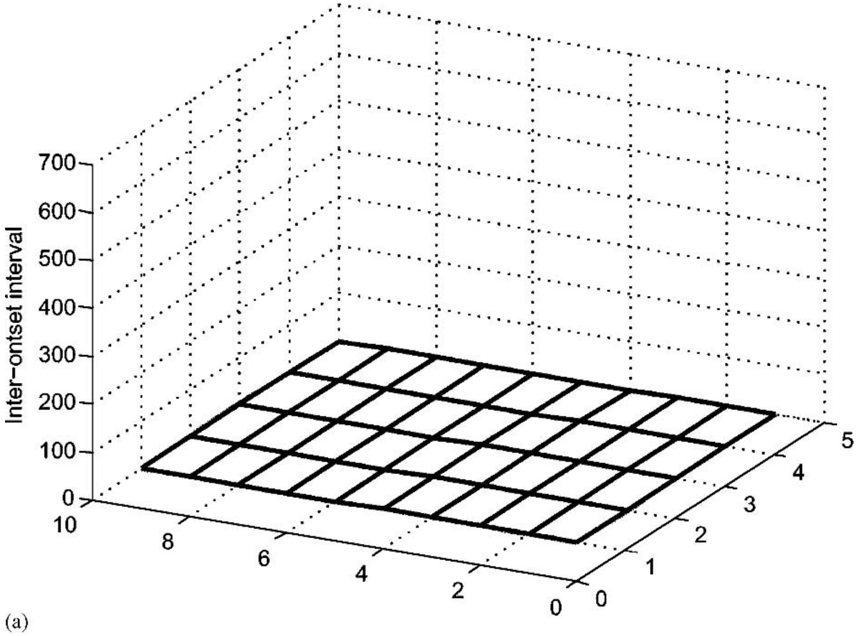


FIGURE 10.14. The evolution of the weights corresponding to IOI without change of neighbourhood: (a) Random initialization of weight values; (b) After 20 iterations; (c) After 80 iterations; (d) Difference between the sum of the weights for consecutive iterations.

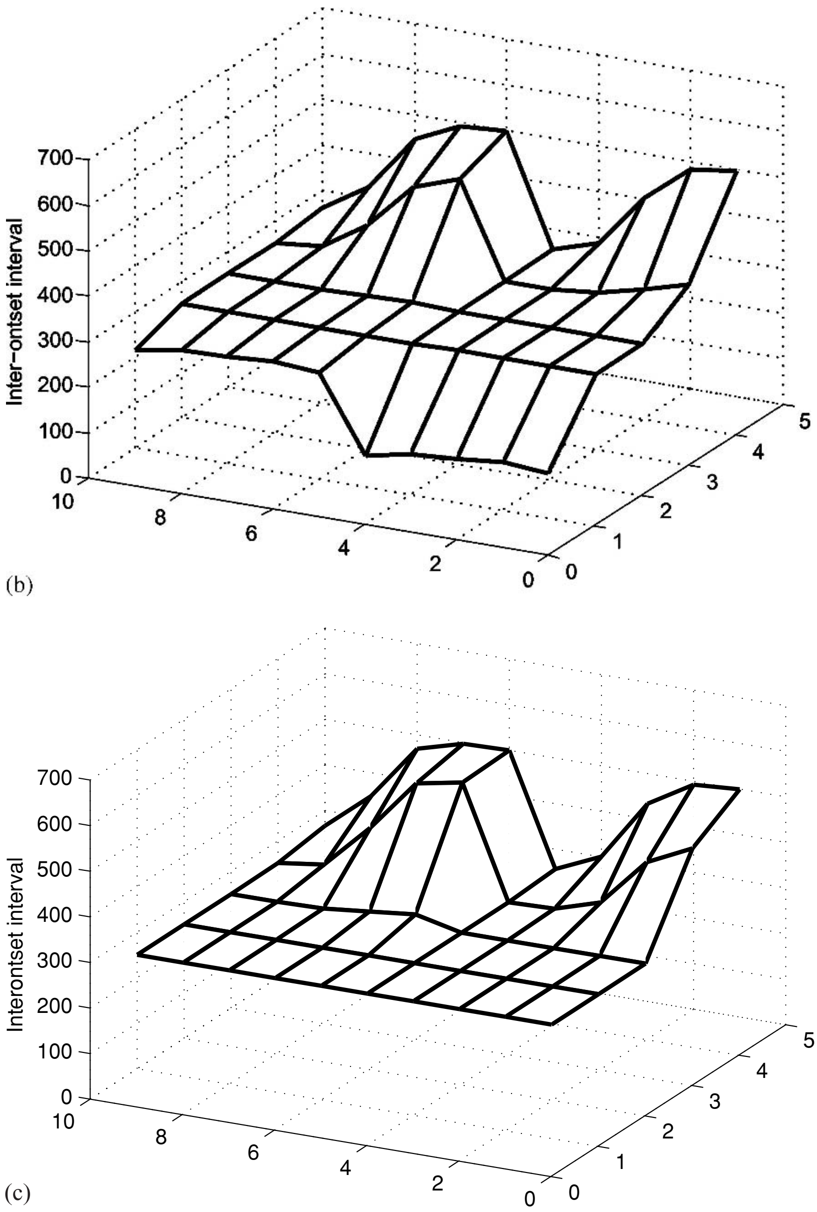
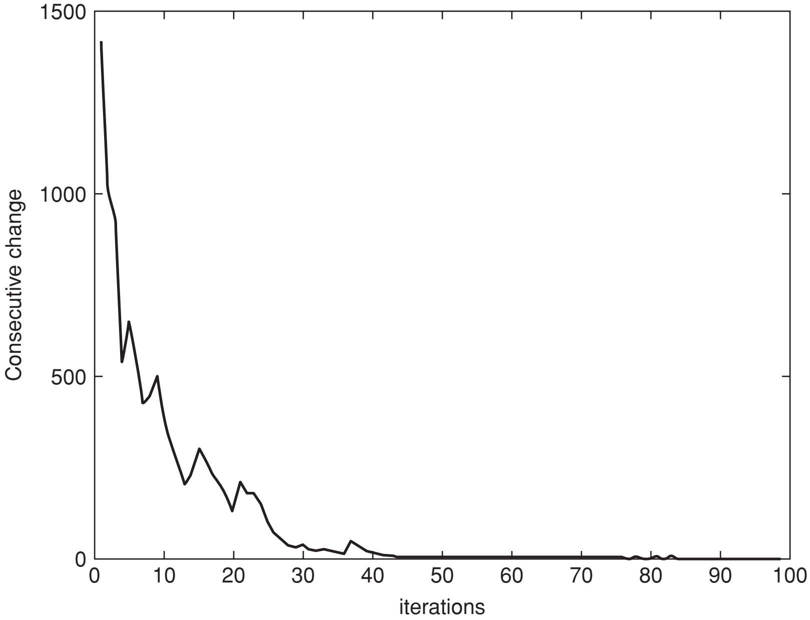


FIGURE 10.14. (*Continued*)

output targets: [0, 0, 1], [1, 0, 0] and [0, 1, 0]. The three learned rhythms are marked with an 'o' in the categorization space (Fig. 10.15). Next, the Sarnet nodes corresponding to the remaining two rhythms were fed into the feedforward network. These are marked with an 'x' in the categorization space. Clearly, the



(d)

FIGURE 10.14. (*Continued*)

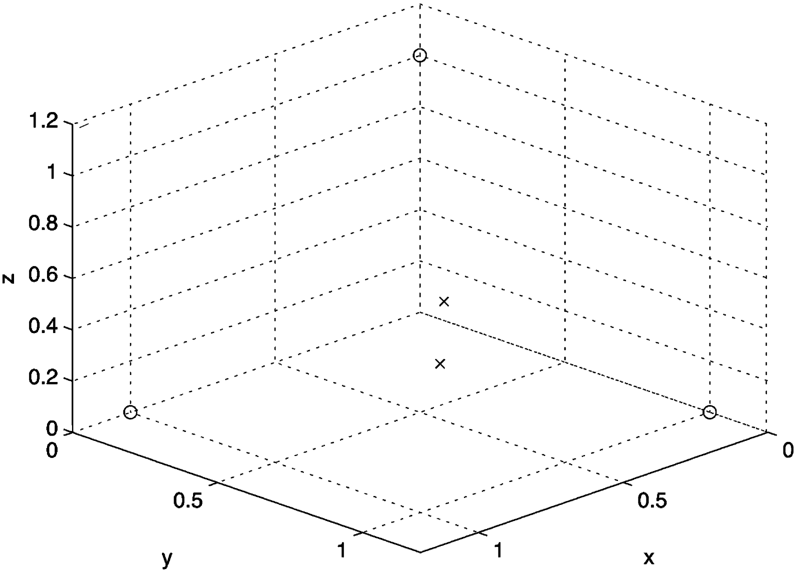


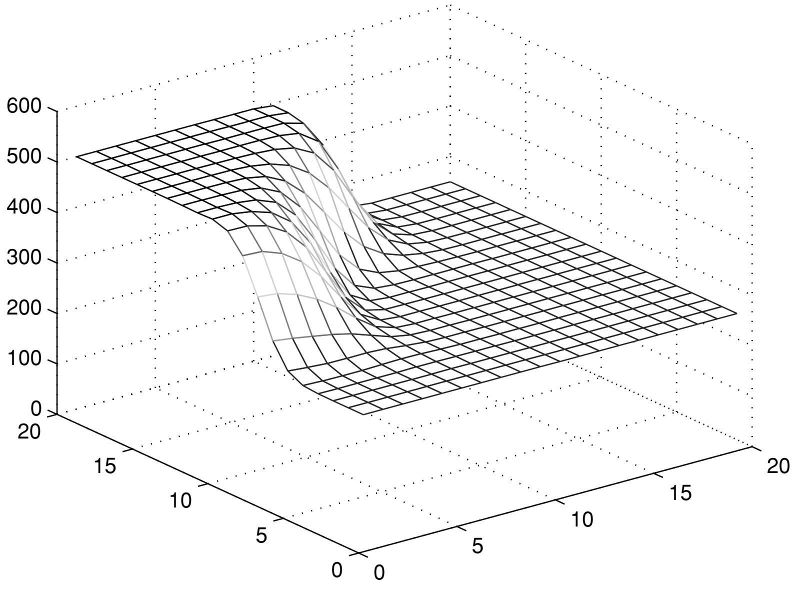
FIGURE 10.15. The tri-dimensional categorization space.

agent is able to distinguish the latter two rhythms from the previous three. This example shows the meaning of the categorization space: For this case the extreme points are the ones acknowledged by the agent as being part of its own repertoire.

As an initial step to test if this architecture would be suitable to be used as a memory mechanism to evolve and store information about sound sequences, two robots were programmed with a Sardnet with 400 (20 × 20) nodes coupled with a feedforward network with 400 input nodes, three nodes in the hidden layer and three output nodes. The choice for number of input nodes is a trade-off between scale and computational weight. The size of the network determines the number of different events that can be encoded, but the larger the network, the slower the computation. The Sardnet was programmed with an initial value for the learning rate $\alpha = 0.08$ and $\sigma = 10$; the latter denotes the scope of the neighbourhood, which is a Gaussian function centered on the winning neuron that multiplies all the elements in the network.

The Sardnet of one of the robots (the robot-player) was trained with eight rhythms (the five ones shown in Fig. 10.13 plus additional three) for 100 iterations. Then its feedforward network was trained with the backpropagation algorithm to respond to these rhythms in the extreme positions of the categorization space. Fig. 10.16(a) depicts the robot-player's pattern of IOI weights, which emerged after the 100 iterations. Fig. 10.17(a) shows the categorization of the rhythms in the tridimensional categorization space.

The other robot (the robot-imitator) was not trained. Its task was to evolve its own rhythmic categorization by imitating the rhythms produced by the robot-player.



(a)

FIGURE 10.16. (a) Robot-player's map of the IOIs; (b) Robot-imitator's map of the IOIs.

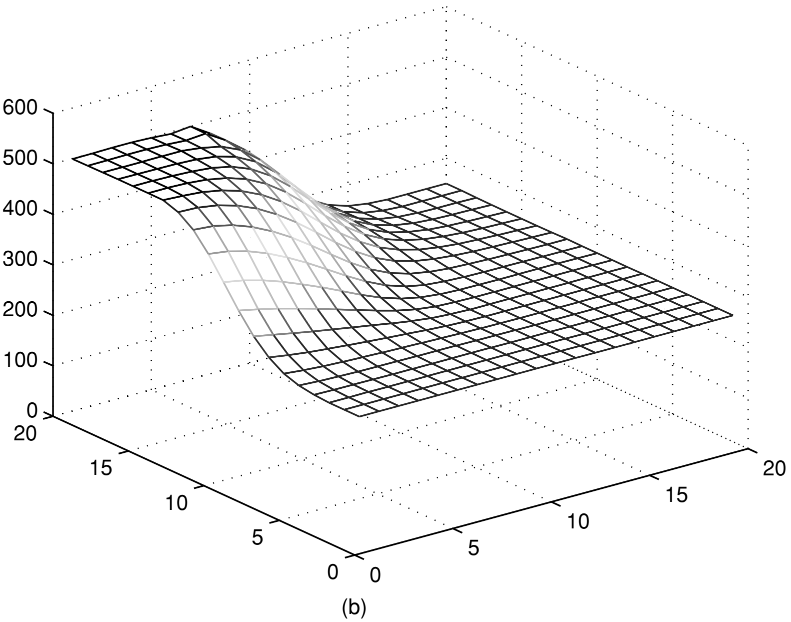
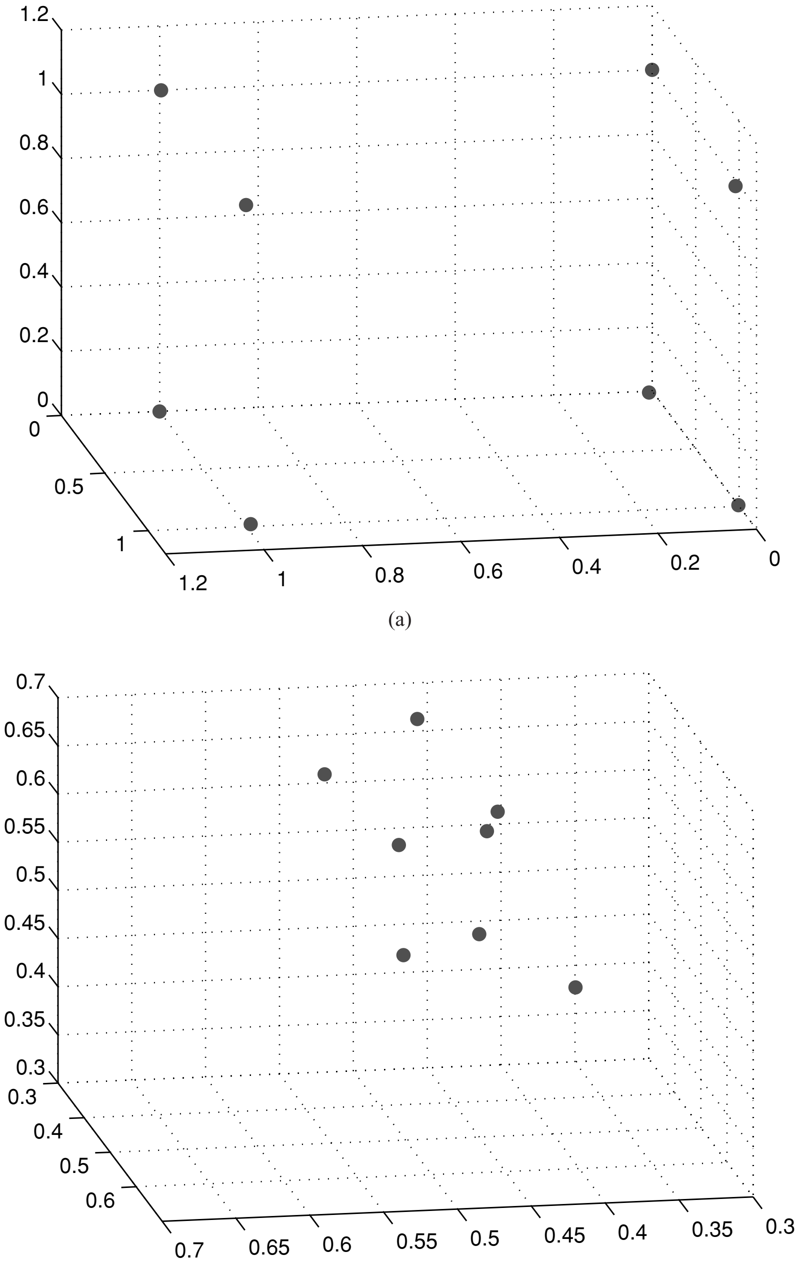


FIGURE 10.16. (*Continued*)

The interaction is as follows: The robot-player plays rhythms randomly picked from its memory and the robot-imitator tries to imitate them. The robot-imitator's learning takes place during this process of imitation. The robot-imitator's neural networks are programmed with the same parameters as the robot-player. Each time the robot-imitator hears a rhythm, it calculates its respective position in its categorization space. Then it 'babbles' a few rhythms (i.e. it generates random rhythms and categorises them) until it finds one that is close enough to the one it that it is trying to imitate. This rhythm is then played as an imitation to the robot-player, who evaluates the imitation in its categorization space. Depending on the distance between the imitation and the original rhythm, the robot-player sends a feedback to the robot-imitator, which indicates two possible outcomes:

a) Satisfactory: The imitation is closer to the original rhythm than to any other rhythm in its categorization space.
b) Unsatisfactory: The imitation is closer to a rhythm other than the original rhythm in its categorization space.

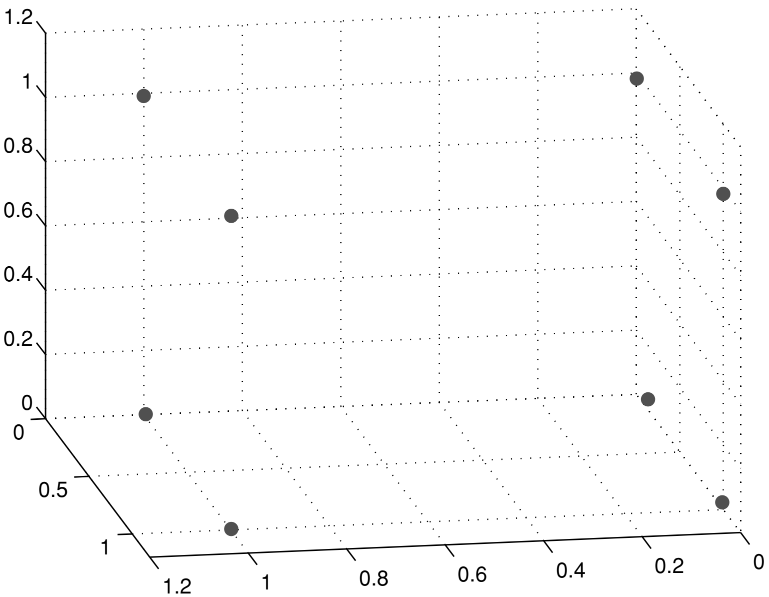
If the imitation is satisfactory, then the feedforward network of the robot-imitator is trained with one iteration of the backpropagation algorithm to respond to the desired category. Conversely, if the imitation is unsatisfactory, then the robot-imitator will benefit only from the adaptation of the Sardnet weights according to its self-organizing behaviour.



(a)

(b)

FIGURE 10.17. (a) The robot-player's trained categories; (b) The categorization map of the robot-imitator at very early stages of the learning process; (c) The categorization map of the robot-imitator at a very late stage of the learning process.



(c)

FIGURE 10.17. (*Continued*)

Fig. 10.16(b) portrays the Sardnet of the robot-imitator after 80 interactions, which is slightly different from the Sardnet of the robot-player, although they seem to be representing the same rhythms.

Fig. 10.17 shows the categorization maps for both robots. Notice that initial categories of the robot-imitator in Fig. 10.17(b) do not match the categories of the robot-player in Fig. 10.17(a), but as the learning process progressed, the categories converged to the extreme points of the tri-dimensional space in Fig. 10.17(c).

A preliminary test of the connectionist memory demonstrated that it is possible to combine the learning capabilities of neural networks with the dynamics of the mimetic model introduced in Section 3. At the time of writing, the connectionist memory is being further improved (e.g. to include information about pitch and timbre) and embedded in the mimetic robotic model.

10.5. Concluding Remarks

This chapter provided a glimpse of the exciting new field of Computational Evolutionary Musicology by demonstrating how music can be studied as an adaptive complex dynamic system. In this context, the origins and evolution of music can be studied using computer models and robotic simulations.

One interesting hypothesis that is emerging from the research of a number of scholars is that there might have been a single precursor for both music and language: A communication system that had the characteristics that are now shared by music and language, but split into two systems at some date in our evolutionary history. For instance, Steven Brown (2000) refers to this single precursor as *musilanguage*, whereas Alison Wray (1998) proposed the notion *holistic protolanguage*, which essentially is the same thing.

In the introduction we suggested that it would be perfectly plausible to suggest that robots might be able to evolve music. Then we demonstrated how this can be done. However, we acknowledge that the examples introduced here cannot evolve proper music yet, but rather the rudiments of what one might refer to as *proto-music*.

Much work is still needed in order to embed the robots with the minimum necessary physiological and cognitive abilities to evolve music. Nevertheless, the models discussed in this chapter are encouraging in the sense that they provide strong indications that music might indeed emerge in a society of interacting autonomous robots.

References

Behr, O. and von Helversen, O. (2004). Bat serenades—Complex courtship songs of the sac-winged bat *Saccopteryx bilineatta*". *Behavioral Ecology and Sociobiology,* **56**: 106–115.

Bidlack, R. (1992). Chaotic systems as simple (but complex) compositional algorithms. *Computer Music Journal*, **16**(3): 33–47.

Boersma, P. (1993). Articulatory Synthesizers for the Simulations of Consonants. *Proceedings of Eurospeech'93*, Berlin, Germany, pp. 1907–1910.

Brodie, R. (1996). *Virus of the Mind: The New Science of the Meme*. Integral Press, Walnut Creek, CA.

Brown, S. (2000). The "Musilanguage" model of music evolution. In N.B. Merker and S. Brown (Eds.), *The Origins of Music*. The MIT Press, Cambridge, USA.

Burton, A.R. and Vladimirova, T. (1997). A Genetic Algorithm Utilising Neural Network Fitness Evaluation for Musical Composition, In G.D. Smith, N.C. Steele and R.F. Albrecht (Eds.), *Proceedings of the 1997 International Conference on Artificial Neural Networks and Genetic Algorithms*, Springer-Verlag, Vienna, pp. 220–224.

Cangelosi, A. and Parisi, D. (Eds.) (2001). *Simulating the Evolution of Language*. Springer Verlag, London, UK.

Casti, J.L. (1997). *Would-be Worlds: How Simulation of Changing the Frontiers of Science*. John Wiley & Sons, NY.

Christiansen, M.H. and Kirby, S. (Eds.) (2003). *Language Evolution: The States of the Art*. Oxford University Press, Oxford, UK.

Cope, D. (1996). *Experiments in Musical Intelligence*. Madison, A-R Editions Inc., WI.

Darwin, C. (1992) (1st published in 1871). *The Descent of Man and Selection in Relation to Sex*. Princeton University Press, Princeton, NJ.

Freeman, W. (1995). *Societies of Brains: A Study in the Neuroscience of Love and Hate*. Lawrence Erlbaum Associates, Mahwah, NJ.

Gallese, V. and Goldman, A. (1998). Mirror-neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences,* **12**: 493–501.

Glaser, R. (2001). *Biophysics*. Springer, Heidelberg.

Holy, T.E. and Guo, Z. (2005). Ultrasonic Songs of Male Mice. *PLoS Biology,* **3**(12): e386.

James, D.L. and Miikkulainen, R. (1995). SARDNET: a self-organizing feature map for sequences. In G. Tesauro, D. Touretzky and T. Leen (Eds), *Advances in Neural Information Processing Systems 7*. MIT Press, Cambridge, MA.

Kohonen, T. (1997). *Self-Organizing Maps*. Springer Series in Information Sciences. Springer-Verlag, Heidelberg.

Levy, S. (1993). *Artificial Life: A Report from the Frontier where Computers meets Biology*. Vintage, London, UK.

Locke, J.L. (1993). *The Child's Path to Spoken Language*. Harvard University Press, Cambridge, MA.

Mâche, F.-B. (1991). *Musique, Mythe, Nature ou les Dauphins d'Arion*. Méridiens Klincksieck, Paris.

Marler, P. and Slabbekoorn, H. (Eds.) (2004). *Nature's music: The science of birdsong*. Elsevier, Boston, MA.

Martins, J. and Miranda, E. R. (2006). A Connectionist architecture for the evolution of rhythms. *Proceedings of EvoWorkshops 2006*, LNCS 3970. Springer, New York, pp. 696–706.

Milicevic, M. (1996). The Impact of Fractals, Chaos and Complexity on Computer Music Composition. *Proceedings of International Computer Music Conference (ICMC 96)*. Hong Kong, International Computer Music Association, San Francisco, pp. 473–476.

Miller, G. (2000). Evolution of human music through sexual selection. In N. Wallin, B. Merker and S. Brown (Eds.), *The Origins of Music*. The MIT Press, Cambridge, MA, pp. 329–360.

Miranda, E. R. and Drouet, E. (2006). Evolution of musical lexicons by babbling robots. *Proceedings of Towards Autonomous and Robotic Systems 2006*, University of Surrey, Gilford, UK. On-line proceedings: http://taros.mech.surrey.ac.uk/schedule.php (Accessed 17 Nov 2006).

Miranda, E.R. (2002b). Mimetic model of intonation. In C. Anagnostopoulou, M. Ferrand and A. Smaill (Eds.), *Music and Artificial Intelligence—Second International Conference ICMAI 2002*. Lecture Notes on Artificial Intelligence 2445, Springer-Verlag, Berlin, Germany, pp. 107–118.

Miranda, E.R. (2002a). *Computer Sound Design: Synthesis Techniques and Programming*. Focal Press, Oxford, UK.

Miranda, E.R. (2001). Synthesising prosody with variable resolution. *AES Convention Paper 5332*. Audio Engineering Society, Inc., NY, USA.

Mithen, S. (2005). *The Singing Neanderthal: The Origins of Music, Language, Mind and Body*. Weidenfeld & Nicolson, London.

Mozer, M. (1994). Neural network music composition by prediction: Exploring the benefits of psychophysical constraints and multiscale processing. *Connection Science,* **6**: 247–280.

Näätänen, R., Tervaniemi, M., Sussman, E., Paavilainen, P. and Winkler, I. (2001). Primitive intelligence in the auditory cortex, *Trends in Neurosciences,* 24: 283–288.

Nazzi, T., Floccia, C. and Bertoncini, J., (1998). Discrimination of pitch contours by neonates. *Infant Behaviour*, **12**: 543–554.

Papadopoulos, G. and Wiggins, G. (1998). A Genetic Algorithm for the Generation of Jazz Melodies. *Proceedings of 8th Finnish Conference on Artificial Intelligence*, Jyväskylä, Finland.

Payne, R.S. and McVay, S. (1971). Songs of humpback whales, *Science,* **173**: 585–597.

Parsons, L.M. (2003). Exploring the Functional Neuroanatomy of Music Performance, Perception and Comprehension, In I. Peretz and R. Zatorre (Eds.), *The Cognitive Neuroscience of Music*. Oxford University Press, Oxford, UK, pp. 247–268.

Peretz, I. and Coltheart, M. (2003). Modularity of music processing. *Nature Neuroscience,* **6**: 688–691.

Peretz, I., Kolinsky, R., Tramo, M., Labrecque, L., Hublet, C. and Demeurisse, G. (1994). Functional dissociations following bilateral lesions of auditory cortex. *Brain,* **117**: 1283–1301.

Rousseau, J.-J. (1990) (1st published in 1765). *Essay sur l'origine des langues*. Gallimard, Paris.

Rizzolatti, G. and Craighero, L. (2004). The mirror-neuron system. *Annual Review of Neuroscience*, **27**: 169–192.

Salu, Y. (2001). *Understanding Brain and Mind: A Connectionist Perspective*. World Scientific, Singapore.

Steedman, M. (1984). A generative grammar for jazz chord sequences. *Music Perception,* **2**: 52–77.

Steels, L. (1997). The Origins of Syntax in Visually Grounded Robotic Agents. *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'97).* Nagoya, Aichi, Japan.

Thomas, D.A. (1995). *Music and the Origins of Language*. Cambridge University Press, Cambridge, UK.

Todd, P.M. and Loy, D.G. (Eds.) (1991). *Music and Connectionism*. The MIT Press, Cambridge, MA.

Todd, P.M. and Werner, G.M. (1999). Frankensteinian Methods for Evolutionary Music Composition. In N. Griffith and P.M. Todd (Eds.), *Musical Networks: Parallel Distributed Perception and Performance*. The MIT Press/Bradford Books, Cambridge, USA, pp. 313–339.

Wallin, N.J., Merker, B. and Brown, S. (Eds.) (2000). *The Origins of Music*. The MIT Press, Cambridge, USA.

Wray, A. (1998). Protolanguage as a holistic system for social interaction. *Language and Communication,* **18**: 46–667.

Zinkovsky, A.V., Sholuha, V.A. and Ivanov, A.A. (1996). *Mathematical Modelling and Computing Simulation of Biomechanical Systems*. World Scientific, Singapore.