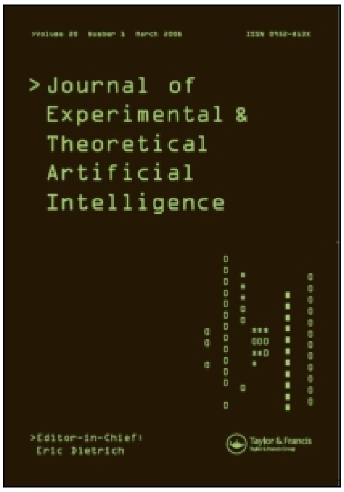


Journal of Experimental & Theoretical Artificial Intelligence
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713723652

Emergent songs by social robots

Eduardo Reck Miranda [a]
[a] Interdisciplinary Centre for Computer Music Research (ICCMR), University of Plymouth, Plymouth, Devon PL4 8AA, UK

First Published on: 15 August 2008

To cite this Article Miranda, Eduardo Reck(2008)'Emergent songs by social robots',Journal of Experimental & Theoretical Artificial Intelligence,

To link to this Article: DOI: 10.1080/09528130701664640
URL: http://dx.doi.org/10.1080/09528130701664640

Journal of Experimental & Theoretical Artificial Intelligence
2008, 1–16, iFirst

Taylor & Francis
Taylor & Francis Group

Emergent songs by social robots

Eduardo Reck Miranda*

Interdisciplinary Centre for Computer Music Research (ICCMR), University of Plymouth, Portland Square Building, B326, Drake Circus, Plymouth, Devon PL4 8AA, UK

(*Received 30 January 2007; final version received 21 November 2007*)

This paper reports the first results of an innovative approach to modelling music cognition based on the emergent behaviour of interacting autonomous systems. A group of interactive autonomous singing robots were programmed to develop a shared repertoire of songs from scratch, after a period of spontaneous creations, adjustments and memory reinforcements. The robots interact with each other by means of vocal-like sounds. They use real sounds as opposed to software simulation. They are furnished with a physical model of the vocal tract, which synthesises vocal singing-like intonations, and a listening mechanism, which extracts pitch sequences from audio signals. The robots learn to imitate each other by babbling heard intonation patterns in order to evolve vectors of motor control parameters to synthesise the imitations. Models of the basic mechanisms underlying the emergence of songs are of great interest for musicians looking for hitherto unexplored ways to create music with interactive machines.

Keywords: experimental AI modelling; emergent communication systems; interactive intelligent systems; learning by imitation; autonomous robots; artificial intelligence and music

1. Introduction

In this paper we introduce an experimental AI system whereby a group of interactive robots programmed with appropriate motor (vocal), auditory and cognitive skills can develop a shared repertoire of short songs from scratch, after a period of spontaneous creations, adjustments and memory reinforcements. The robots develop vectors of motor control parameters to produce imitations of heard songs.

Why is it useful to model the emergence of repertoires of songs with a robotic model? A better understanding of basic mechanisms underlying the emergence of song patterns in an artificial system is of great interest for musicians looking for hitherto unexplored ways to create music with interactive intelligent machines. Broadly speaking, current AI

*Email: eduardo.miranda@plymouth.ac.uk

ISSN 0952–813X print/ISSN 1362–3079 online

DOI: 10.1080/09528130701664640
http://www.informaworld.com

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 Cellular Automata (Hunt, Orton and Kirk 1991, Miranda 1993) and Particle Swarms (Blackwell and Bentley 2002) 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 machine learning 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. The goal of our research is to bring the merits of both approaches closer to each other. Music consists of units organised in specific ways and it is culturally transmitted. Models of the mechanisms underlying the dynamics of such organisation and cultural transmission are bound to provide new insights into building interactive intelligent music systems.

2. The model

The robots in a group are expected to form a common repertoire of songs: a robot must develop a repertoire that is similar to the repertoires of its peers. Metaphorically speaking we could say that the songs create some form of 'social identity' for the robots, which can be assessed in terms of the similarity of their repertoires.

The importance of imitation for the acquisition of behaviour 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).

2.1 *The architecture*

The robots (Figure 1) are equipped with a voice synthesiser, a hearing apparatus and a memory device.

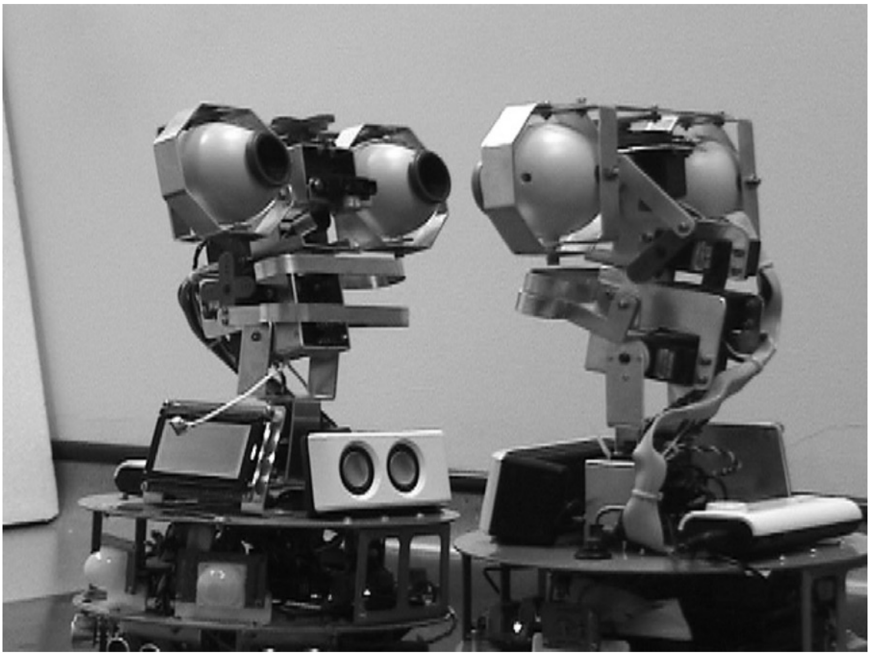


Figure 1. The model uses commercially available robots, which were adapted at ICCMR for high-quality voice synthesis and analysis with sampling rate at 22 050 Hz.

The voice synthesiser is implemented as a physical model of the vocal tract, which is able synthesise formants and a number of vocal-like sounds. The robots need to compute three vectors of parameters for the synthesiser in order to produce vocal-like intonations: (a) lung pressure, (b) the width of the glottis, and (c) the length and tension of the vocal chords, represented as lung_pressure(n), interarytenoid(n) and cricothyroid(n), respectively (Boersma 1993, Miranda 2002). 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. For the sake of consistency, from now on the term ‘intonation’ will be used instead ‘song’.

Essentially, the memory of a robot stores its repertoire of intonations, but it also stores other information such as probabilities, thresholds and reinforcement parameters. (These variables will be clarified when the algorithms are introduced below. See also the Appendix.) The robots 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, which is represented using a representation scheme of our own design: CARMEC (Common Abstract Representation of Melodic Contour). CARMEC represents a perceptual map 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 (Figure 2):

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

where:

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

and

VLSU = very large step up
LSU = large step up

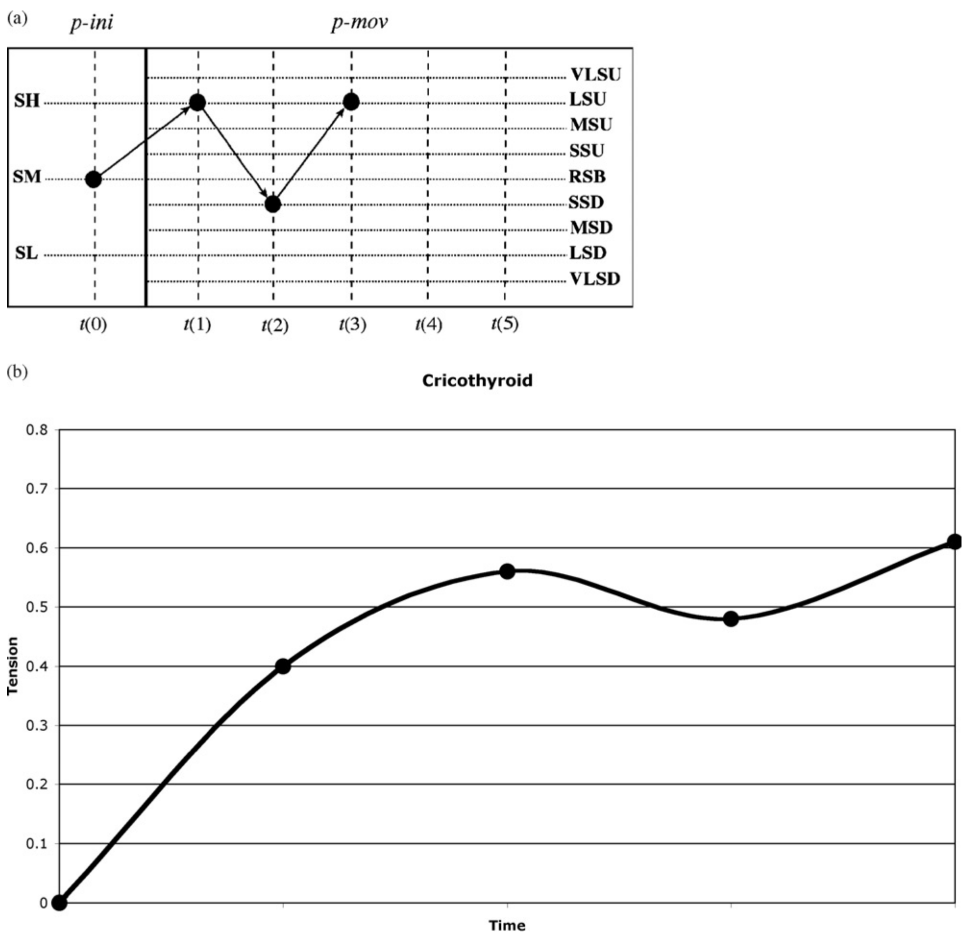


Figure 2. (a) The representation of an intonation, where $t(n)$ indicates an ordered sequence of n pitches. (b) The cricothyroid vector of the motor control map that produced this intonation.

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 pitch frequency values or labels for musical notes are not relevant here because the objective is to represent abstract melodic contours rather than a sequence of pitches (or musical notes) drawn from a specific tuning system. This is very important because one should not assume that the robots must sing in any pre-established musical scale. Rather, they should be given the ability to establish their own tuning system collectively.

2.2 *The interactions*

The interaction algorithms were largely inspired by the work of Steels (1997) on 'evolutionary' language games. A similar interaction dynamics has been proposed by de Boer (2001) to implement a model for the development of vowel systems.

All robots have identical synthesis and listening apparatus. At each round, each of the robots in a pair plays one of two different roles: the robot-player and the robot-imitator.

The main algorithms are given in detail in the Appendix. Glimpses at the functioning of these algorithms are given in Figures 3, 4 and 5. For didactic purposes, the co-ordinates of these figures do not correspond to the actual parameters of the model. For the sake of clarity, the plotting is in an idealised two-dimensional representation of the motor and perceptual repertoires. The numbers in the figures indicate actions corresponding to the line numbers of the algorithms in the Appendix. The robots do not sing all at the same time; they interact in pairs.

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. Figure 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). (Note: the numbers in parenthesis correspond to actions represented by dotted arrows in Figures 3, 4 and 5.) 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 repertoire the intonation ϕ that is most perceptually similar to the imitation ψ (10).

If the robot-player finds another intonation ϕ that is closer to Δ than α is, then the imitation is seen as unsatisfactory, otherwise it is satisfactory. In Figure 3, the robot-player

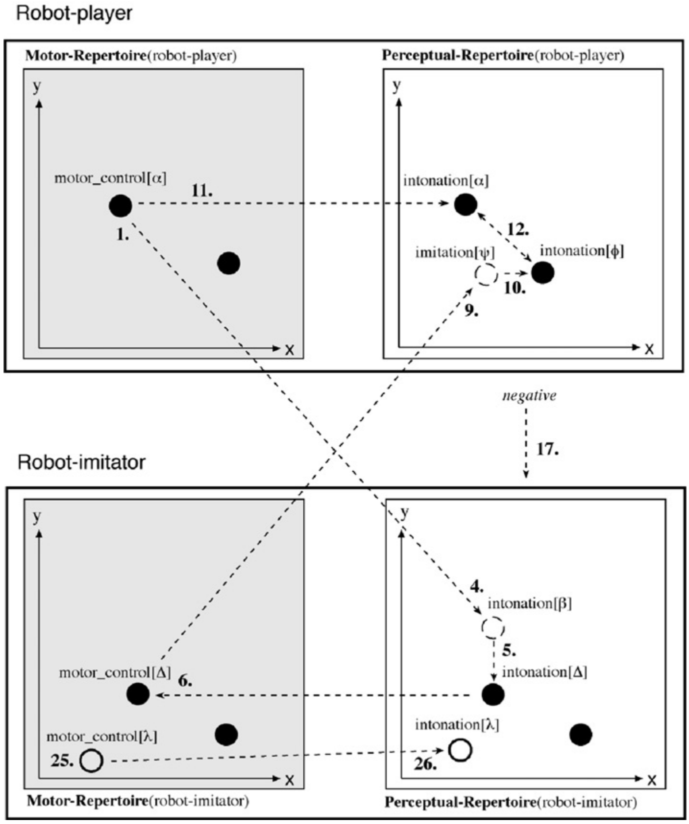


Figure 3. Example of an unsuccessful imitation.

babbles the original intonation α to itself (11) and concludes that α and ϕ are different (12). Then, it sends a negative feedback to the robot-imitator (17). 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 slightly from α (by means of a deviation coefficient), 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 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 intonations to itself (alike 'babbling') and then it picks the one that is perceptually most similar to Δ. Let us assume that in Figure 3 the intonation Δ 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).

Figure 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, Figure 5 shows what would have happened 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

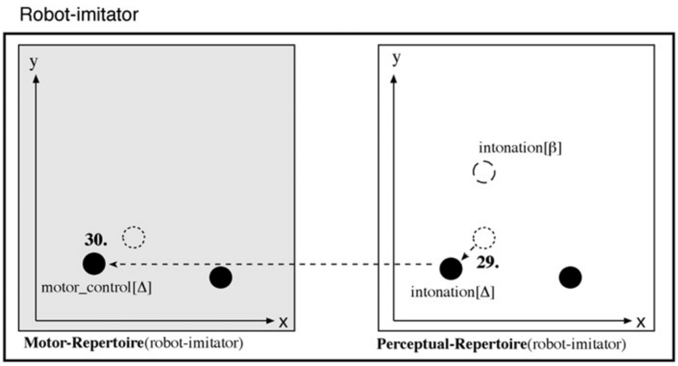


Figure 4. An example where the unsuccessful imitation involved an intonation that has a poor past success rate.

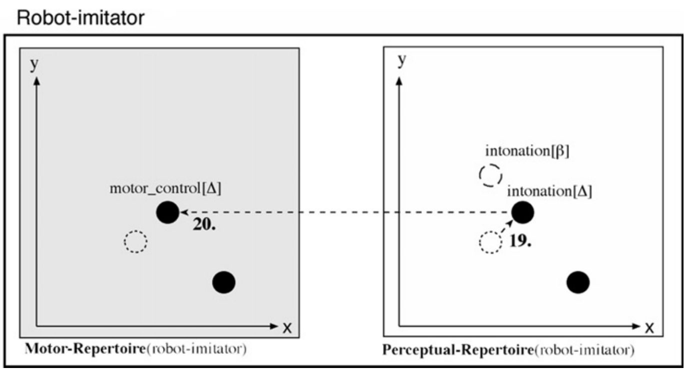


Figure 5. An example of a successful imitation.

memory and would have moved it slightly towards the representation of the heard intonation β.

Before terminating the round, both robots perform final updates. First, 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'.

The signal feedback is implemented as follows: if feedback is positive, then the robot makes a couple up-and-down movements of its head. If negative, then it makes a couple of left-to-right movement of its head.

3. The behaviour of the model

Although we can run the model in simulation mode with software agents, with the exception of the last example in Section 3.4, all examples discussed below are from robotic interactions with real sounds. However, due to limited resources, we have used only two robots, which provided embodiment to pairs of software agents. The agents were transmitted wireless to the robots from a computer and vice-versa. We also included a band-pass filter in the listening system tuned to the range of frequencies of the sounds they can possibly sing; this is to avoid the interference of the noise generated by the motor that moves the robot's mouth.

Figure 6 plots an example of an intonation with three elements in the sequence and its respective pitch analysis and CARMEC representation. The oscillations of the line representing pitch in Figure 6b are due to the vibrato nature of the singing voice, which in this case is a modulation of approximately 5 Hz.

3.1 *Average size of the evolved repertoire*

The graph in Figure 7 shows a typical example of the development of the average repertoire of a group of five robots, with snapshots taken after every 100 interactions over a total of 5000 interactions. The robots developed repertoires averaging 19 intonations each. (Note that some may have developed more or less than 12 intonations.) After a steady steep increase of the repertoire until about 2000 interactions, the robots settled to an average of 17 intonations each until about 2400 interactions. Then they increased the repertoire to an average of 18 intonations until about 4000 interactions. From 4000 interactions onwards the robots finally settled to an average of 19 intonations.

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 effect of running the simulation with a larger group and for longer is shown in Section 3.4.

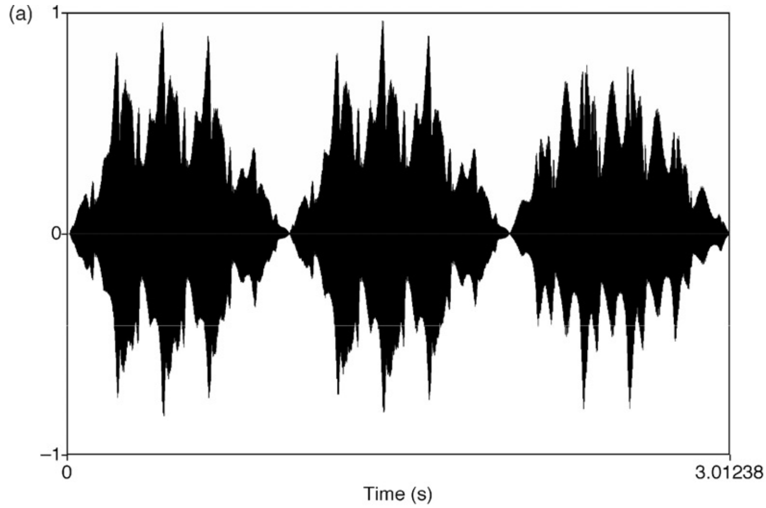
Figure 8 portrays the perceptual memory of a robot randomly selected from the group after 5000 interactions. In this case, the length of the intonations varied from three to six pitches. (The minimum and maximum length of the intonation to be evolved is fixed beforehand.) This robot evolved 11 intonations; one below the average of the group.

3.2 *Rate of successful imitations*

The graph in Figure 9 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 regain the 100% success rate stability.

3.3 *Perceptual vs. motor maps*

An interesting feature of this model is that the robots do not necessarily have to evolve the same motor representations for what is considered to be perceptibly identical. Figure 10 shows the motor functions (Figures 10b, 10c and 10d) evolved by three different robots



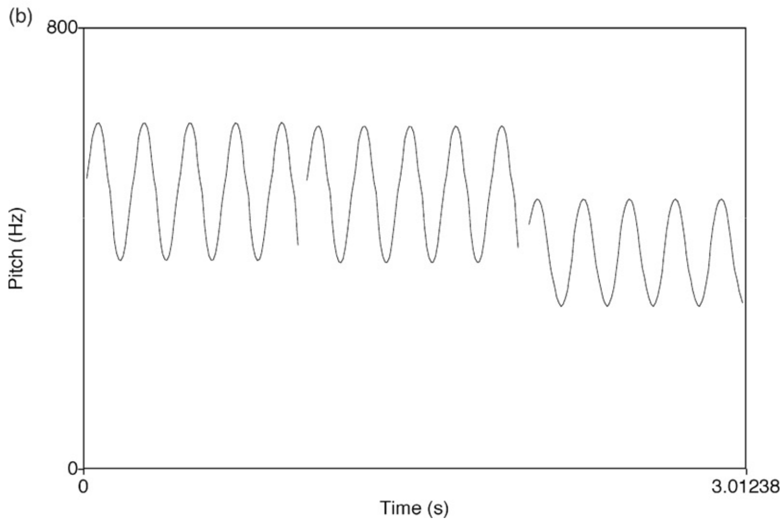


Figure 6. An example of an intonation produced by a robot. (a) Plotting of the intonation. (b) The pitch analysis of the intonation. (c) The CARMEC representation: the intonation starts at the higher register (SH), then it remains within the same frequency range (RSB), and finally it steps down slightly (SSD). For the sake of clarity, the background metrics and labels of the graph in 6(c) are not shown. Please refer to Section 2.1, Figure 2 for an explanation of the perceptual representation.

(Robot 1, Robot 2 and Robot 3) to represent what is essentially the same intonation (Figure 10a).

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 recognise everything they hear because in order to produce an imitation a robot will use the motor vectors that best match its perception of the intonation in question. It is the robot-player who will assess the imitation.

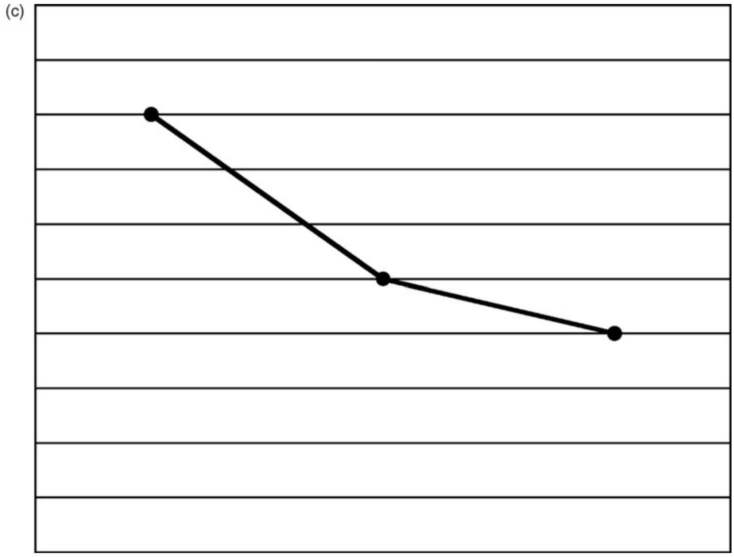


Figure 6. Continued.

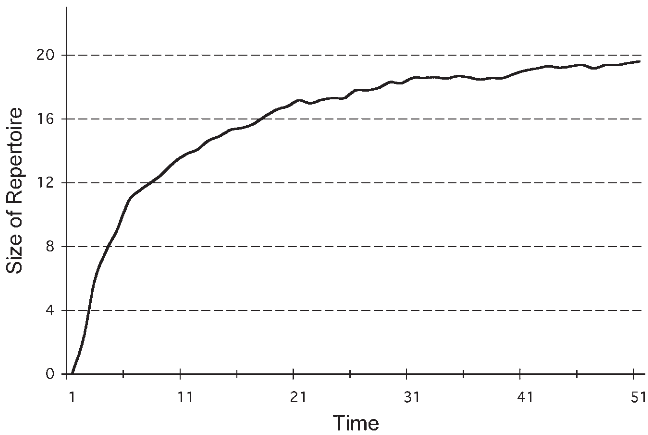


Figure 7. The evolution of the average sise of the repertoire of intonations of the whole group of robots. In this case the group developed an average repertoire of 19 intonations. (The time axis is in terms number of interactions multiplied by 100.)

3.4 *Effect of scaling up*

The graph in Figure 11 shows an example of the development of the average repertoire of a group of 20 robots, with snapshots taken every 100 interactions over a total of 40 000 interactions. In this case the group developed an average repertoire of just over 21 intonations.

This example demonstrates the fact that, given the appropriate simulation settings, the size of the repertoire of intonations tends to stabilise with time. This happens because the

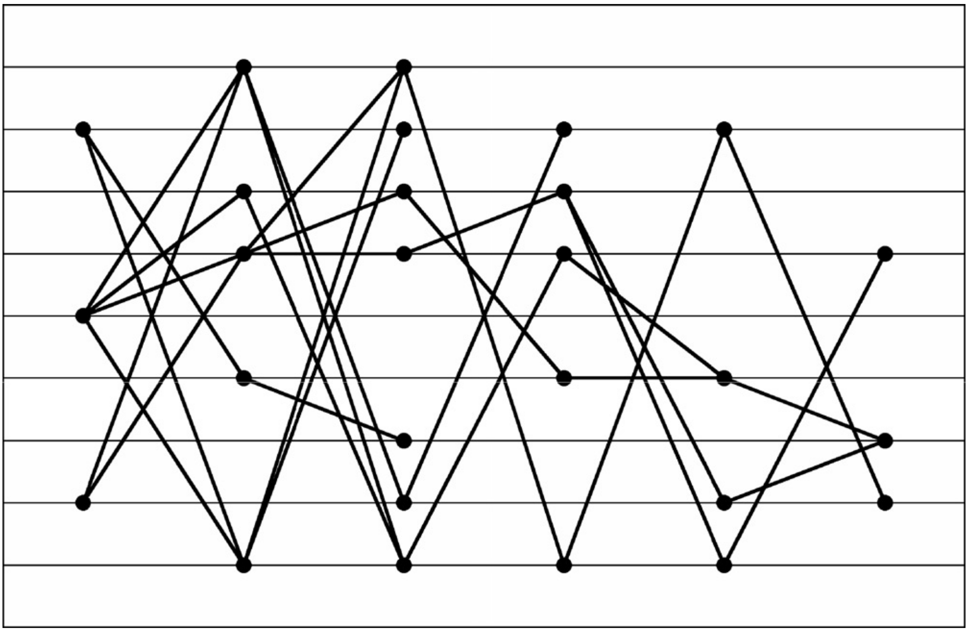


Figure 8. The perceptual memory of one robot. For the sake of clarity, the background metrics and labels of the perceptual representation are not shown. Please refer to Section 2.1, Figure 2 for an explanation of the perceptual representation.

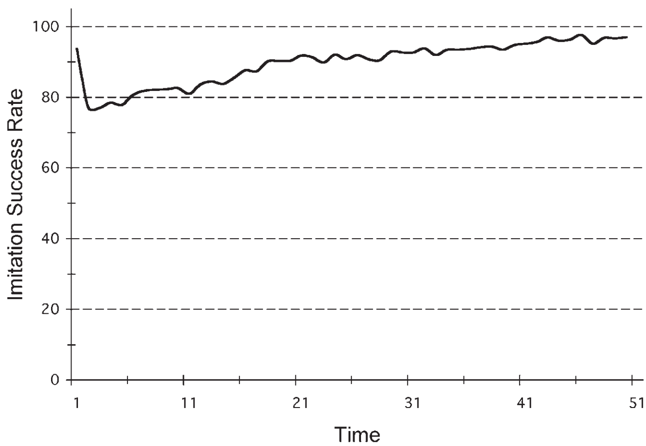


Figure 9. The imitation success rate over time. (The time axis is in terms of number of interactions multiplied by 100.)

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. But these repertoires are dynamic in the sense that these strongly settled intonations do not remain static. The robots are constantly making small adjustments to their representations.

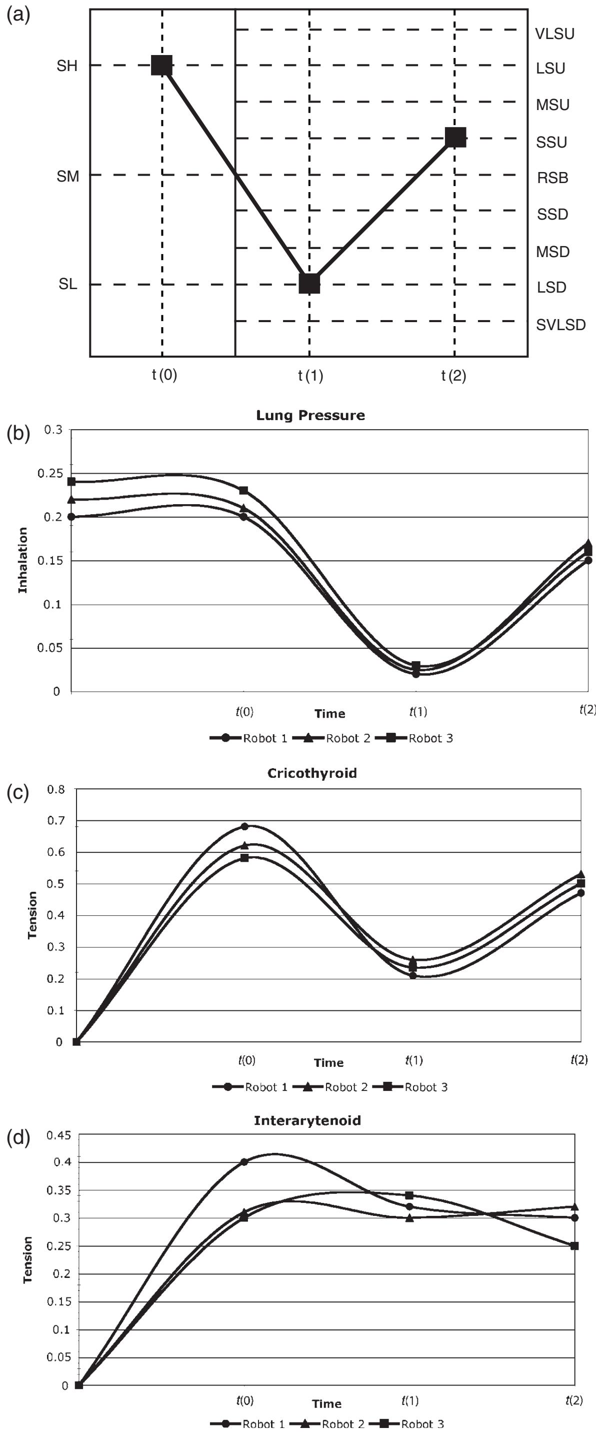


Figure 10. (a) An example of a perceptual pattern lasting for three pitches and its corresponding motor control vectors developed by three different robots: (b) the lung_pressure vector, (c) the cricothyroid and (d) the interarytenoid vector.

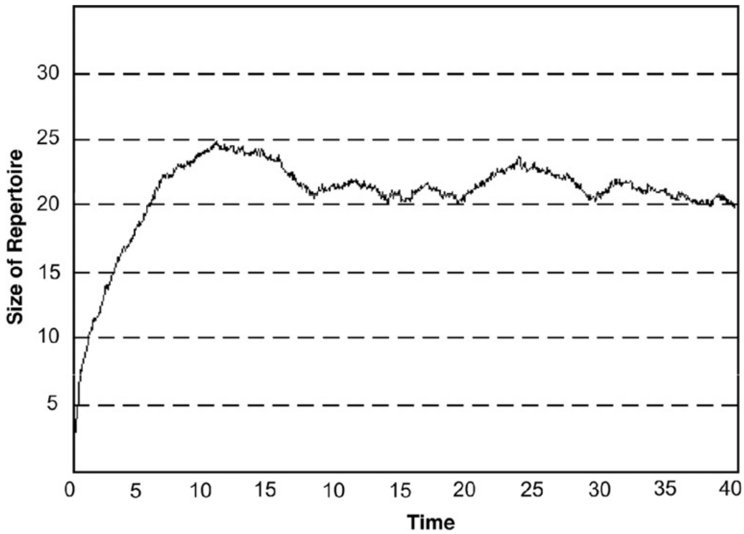


Figure 11. The evolution of the sise of the repertoire of intonation of a group of 20 robots during 40 000 interactions. In this case the group developed an average repertoire of just over 21 intonations. (The time axis is in terms of interactions multiplied by 1000.)

The scaling up was studied by means of a simulation using software agents rather than robots. Considering that each robotic interaction takes an average of 30 seconds for intonations ranging from three to six pitches, a run with 40 000 interactions would take approximately 2 weeks to complete. Software simulation is desirable in such circumstances to fine-tune the model prior to running the robotic simulation.

4. Conclusion and further work

At the core of our the system introduced in this paper is a selective mechanism inspired by Neo-Darwinian evolutionary theory (Fisher 1930, Haldane 1932, Huxley 1942), whereby some form of 'mutation' takes place (e.g. intonations move closer to or away from other intonations in memory) and patterns are 'born' (e.g. with random additions through the 'creativity coefficient') and 'die' (e.g. the spring-cleaning mechanism).

At the introduction we suggested that models such as the one presented in this paper have the potential to shed new insights into building interactive music systems. What sort of systems can be built informed by such models?

We are aiming at the development of technology for implementing intelligent systems that can improvise music in real-time with human musicians. However, instead of manually programming these machines with prescribed rules for generating music, we aim at programming them with the ability of developing these rules autonomously, dynamically and interactively. This paper demonstrated one of the various intertwined algorithms that may lead to such capability.

In order to reduce ambient noise interference, we ran the experiments in a quiet room. However, this constraint will be lifted in further experiments in order to allow for simulations where ambient noise will be taken into account, including interaction with humans singing and/or playing musical instruments. However, before we let the robots interact with humans there are a number of issues to be addressed in order to increase the sophistication of the model.

A limiting aspect of the present system is that the robots only deal with short pitch sequences. The natural progression in this research is to furnish the robots with the ability to deal with longer pitch sequences, rhythm and other musical attributes.

Although the symbolic sensory-motor-like memory mechanism developed for storing intonations served the present model well, it is not efficient for storing longer pitch sequences, let alone other musical attributes. 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. We are looking into the possibility of doing this by means of algorithms for the evolution of grammars (Miranda et al. 2003) and neural networks that mimic the behaviour of mirror neurons (Westerman and Miranda 2003).

Also, we are currently considering ways in which to embed the robots with more sophisticated physiological and cognitive abilities. Although embodiment is not crucial for the present implementation, we have adopted the embodied artificial intelligence approach from the outset of this research because embodiment is a vital part of the types of behaviour that we are hoping to observe in the future; such as, for example, the role of gesture in the development of music. We are currently looking into the implementation of a more sophisticated moving lips mechanism, which will be sensed by the visual system of the imitating robot, thus adding more information for cognition. Information from other sensors will also be used to provide greater immersion of the robots in the environment. Eventually, we will have to move to more morphologically apt robotic platforms (e.g. with a mechanical vocal tract or robots with arms and hands to play instruments, etc.) so as to maximise the roles of embodiment and morphology in the development of behaviour.

Acknowledgements

The author would like to thank the evolutionary music team at ICCMR for their contribution to this research, particularly João Martins and Etienne Drouet, who took active part in the implementation of the robotic simulations. Also, many thanks to Mark Witkowski of the Department of Computing, Imperial College London, for his useful comments on an early version of this paper. Many thanks to Springer Science and Business Media for the kind permission to reprint the figures that appeared originally in the author's book *Evolutionary Computer Music*, 2007.

References

Blackwell, T.M., and Bentley, P.J. (2002). 'Improvised Music with Swarms,' in *Proceedings of the 2002 Congress On Evolutionary Computation*, pp. 462–1467.

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

Burton, A.R. and Vladimirova, T. (1997). 'A Genetic Algorithm Utilising Neural Network Fitness Evaluation for Musical Composition,' in *Proceedings of the 1997 International Conference on*

Artificial Neural Networks and Genetic Algorithms, eds. G.D. Smith, N.C. Steele and R.F. Albrecht, pp. 220–224.
Cope, D. (1996). *Experiments in Musical Intelligence*, Middleton: A-R Editions.
de Boer, B. (2001). *The Origins of Vowel Systems*, Oxford: Oxford University Press.
Fisher, R.A. (1930). *The Genetical Theory of Natural Selection*, Oxford: Clarendon Press.
Haldane, J.B.S. (1932). *The Causes of Evolution*, London: Longman Green.
Hunt, A., Orton, R. and Kirk, R. (1991). 'Musical Applications for a Cellular Automata Music Workstation,' in *Proceedings of the International Computer Music Conference (ICMC 91)*, pp. 165–168.
Huxley, J.S. (1942). *Evolution: The Modern Synthesis*, London: Allen and Unwin.
Locke, J.L. (1993). *The Child's Path to Spoken Language*, Cambridge: Harvard University Press.
Miranda, E.R. (2002). *Computer Sound Design: Synthesis Techniques and Programming*, Oxford: Elsevier/Focal Press.
Miranda, E.R. (2001). 'Synthesising Prosody with Variable Resolution,' in *AES Convention Paper 5332*, New York: Audio Engineering Society.
Miranda, E.R. (1993). 'Cellular Automata Music: An Interdisciplinary Project,' *Interface*, 22, 3–21.
Miranda, E.R., Kirby, S., and Todd, P. (2003). 'On Computational Models of the Evolution of Music: From the Origins of Musical Taste to the Emergence of Grammars,' *Contemporary Music Review*, 22, 91–111.
Mozer, M. (1994). 'Neural Network Music Composition by Prediction: Exploring the Benefits of Psychophysical Constraints and Multiscale Processing,' *Connection Science*, 6, 247–280.
Papadopoulos, G. and Wiggins, G. (1998). 'A Genetic Algorithm for the Generation of Jazz Melodies,' in *Proceedings of the 8th Finnish Conference on Artificial Intelligence*.
Rizzolatti, G., and Craighero, L. (2004). 'The Mirror-Neuron System,' *Annual Review of Neuroscience*, 27, 169–192.
Robinson, F.N. (1975). 'Vocal mimicry and the evolution of bird song,' *Emu – Austral Ornithology*, 75, 23–27.
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,' in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'97)*.
P.M. Todd, and D.G. Loy, (eds.) (1991), *Music and Connectionism*, Cambridge: The MIT Press.
Westerman, G., and Miranda, E.R. (2003). 'Modelling the Development of Mirror Neurons for Auditory-Motor Intergration,' *Journal of New Music Research*, 31, 367–375.

Appendix: The Interaction Algorithms and Settings

Algorithm 1: Robot-player produces an intonation

1. motor_control[α] $\Leftarrow$ pick-any-motor-control in **Motor-Repertoire**(robot-player)
2. synthesise-sound(motor_control[α])

Algorithm 2: Robot-imitator produces an imitation

3. pitch_vector[β] $\Leftarrow$ perceive-intonation
4. intonation[β] $\Leftarrow$ perceptual representation(pitch_vector[β])
5. intonation[Δ] $\Leftarrow$ search-similar(intonation[β]) in **Perceptual-Repertoire**(robot-imitator)
6. motor_control[Δ] $\Leftarrow$ retrieve_motor_control (motor-control[intonation[Δ])
7. synthesise-sound(motor_control[Δ])

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

```
pitch_vector[ψ] ⇐ perceive-imitation
imitation[ψ] ⇐ perceptual-representation(picth_vector[ψ])
intonation[ϕ] ⇐ search-similar(imitation[ψ]) in Perceptual-Repertoire(robot-imitator)
intonation[α] = perceptual-representation(motor_control[α])
IF intonation[α] = intonation[ϕ]
   THEN {feeback ⇐ positive
         reinforce(motor_control[α] in Motor-Repertoire(robot-player)
         reinforce(intonation[α]) in Perceptual-Repertoire(robot-player)}
   ELSE {feeback ⇐ negative}
output-signal(feedback)
```

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

```
IF feedback = positive
   THEN {approximate(intonation[Δ] → intonation[β]) in Perceptual-Repertoire
   (robot-imitator)
      reconfigure_motor_control(intonation[Δ]) in Motor-Repertoire(robot-imitator)
      reinforce intonation[Δ] in Perceptual-Repertoire(robot-imitator)
      reinforce motor_control(Δ) in Motor-Repertoire(robot-imitator)}
   ELSE IF feedback = negative
      THEN IF success-history(intonation[Δ]) > success-threshold
         THEN {motor_control[λ] ⇐ produce-new-motor-control
         intonation[λ] ⇐ perceptual representation(motor_control[λ])
         save-new(intonation[λ]) to Motor-Repertoire(robot-imitator)
         save-new(motor_control[λ]) to Perceptual-Repertoire(robot-imitator)}
      ELSE {distantiate(intonation[Δ] ↔ intonation[β]) in Perceptual-Repertoire
      (robot-imitator)
            reconfigure_motor_control(intonation[Δ]) in Motor-Repertoire(robot-
            imitator)}
```

Algorithm 5: End of interaction updates

```
interaction-updates(robot-player)
interaction-updates(robot-imitator)
```

The frequency of the intonations ranges from 200 Hz to 800 Hz. The robots start every simulation with one sound in their repertoire. Before terminating a round, both robots perform some updates. Firstly they scan their repertoire and merge those sounds that are considered to be perceptibly close to each other. 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 sounds; if a sound has been used more E_T times and has scored less than H% of its total use, then it is deleted. Finally, at the end of each round, the robot-imitator has a certain probability P_a, the 'creativity coefficient', of adding a new randomly created sound to its repertoire. In order to replicate the simulations reported in this paper, the following settings may be used:

Creativity coefficient: P_a = probability of 1%
Forgetfulness disposition: P_b = probability of 20%
Success History threshold: H = 90% of the total number of times a given intonation was used
Erase threshold: $E_T = 50$