This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Towards A Learning Framework for Dancing Robots

Ibrahim S. Tholley, Qinggang Meng & Paul. W. H. Chung

Computer Science Department and Research School of Informatics, Loughborough University, UK

Abstract— How can we make robots learn how to dance? How do humans learn to dance? An emerging culture of dancing robots is becoming more prominent in the research community with more emphasis on how we can show of our own creativity rather than allowing the robots to develop their own cognitive and psychological behaviours to the music being played. There are many different types of music and indeed, many different robots and many ways, in which they can dance to music however, much of the work carried out in this field concern limiting robots to dance in particular ways to a specific music and no adaptive behaviour implemented in them to be able to respond intuitively to music in general. We propose in this paper, a way in which such a problem can begin to be looked into, by introducing fundamental things that should be learnt that are necessary for dancing. We programmed a virtual robot to learn to dance to the beat as well as recognise the downbeat of any time-signature and tailor its movements to the loudness of music, using the Sarsa and the Sarsa(λ) algorithms from reinforcement learning as the learning framework. Experimental results show that it is possible to make robots learn to dance to these fundamental rhythmic features of music.

I. INTRODUCTION

DANCING is a social and entertaining activity for humans, which comes natural [13] to humans. It is a complex non-verbal way in which we communicate with each other and express ourselves and the music being played. The underlining problem is how such an activity can be applied to robots. Making robots dance is a growing research interest. Besides the intrinsic research challenges, the main purpose of dancing robots, is entertainment. Robots are either preprogrammed with dance sequences or dance actions to specific music, but the problem is that they cannot respond adaptively to different types of music.

The way we dance is largely due to how we perceive the music. This is expressed through our movements. Our dancing is influenced by how others perceive our own movements and our own judgments, however, there are certain dancing rules that perhaps we can all agree on, for example, one way in which we assess a person's dancing is by their ability to correctly map their movements to the beat, often carried out by the single continuous movement of the

head or the tapping of the feet. From psychology, it is this synchronisation of the movement to the perceived rhythmic pattern (e.g. the beat) of the music that acts as an intrinsic reinforcement to the dancer, which encourages dancers to continue dancing to the beat.

Of course, this is not always a simple task since there are some music that have complicated beats within them. As part of this ongoing research into how we can make robots dance, we first begin to address this problem by presenting in this paper a reinforcement learning approach to make a virtual robot respond rhythmically to the music. We use the traditional Sarsa and Sarsa(λ) algorithms from the reinforcement learning (RL) framework and the Softmax action-selection method [16] to show how effectively, robots can learn to dance.

The rest of this paper is organised as follows: Section II presents some related material relevant to this research; Section III discusses the implementation of the RL algorithms in three experiments; Section IV presents the theoretical experimental results of the experiments; Sections V and VI conclude this paper with discussions and future work.

II. RELATED WORK

A. Dance Synchronisation and Timing

Dance is an emotional expression to music and is performed by both humans and animals to express their physical, cognitive, and affective domains [12] of the music. It is also a social activity in which some may consider it to be a form of comfort, exercise or self-expression, which gives us the “feel-good factor” either watching it or participating in it, and provides room for judgment and criticisms.

As a measure of time-keeping and staying in rhythm, a typical activity of humans is the tapping of the feet or the moving of the head, which coincide (or is *in-phase*) with the perceived beats of the music [13]. This helps us to get a *feel* of the music and get an understanding of its structure so that we can later explore other movements. On-beat tapping is generally easier to achieve than off-beat tapping [2][13] (unless you are musically trained) and so people tend to respond to on-beat rhythms in their dancing because it is more measurable and perhaps, more intrinsically rewarding. However, if we observe someone learning to dance or trying to learn a new dance, the movements may regularly be perceived as *off-beat* and thus, not in sync with the music. But even off-beat dancing can be in sync with the music and in fact it is what dance experts say makes a “good dance”.

Timing is also important, to know when to come in to the

Manuscript received April 15, 2009. This work was supported by the Computer Science Department and Research School of Informatics of Loughborough University.

I. S. Tholley is with the Computer Science Department and Research School of Informatics, Loughborough University, Leicestershire, UK, LE11 3TU (e-mail: I.Tholley@lboro.ac.uk).

Q. Meng is with the Computer Science Department and Research School of Informatics, Loughborough University, Leicestershire, UK, LE11 3TU (e-mail: Q.Meng@lboro.ac.uk).

P. W. H. Chung is with the Computer Science Department and Research School of Informatics, Loughborough University, Leicestershire, UK, LE11 3TU (e-mail: P.W.H.Chung@lboro.ac.uk).

music with a new movement, and is a matter of good judgment with the music. Fortunately, the music itself provides some assistance to this with fundamental features such as rhythm, and dynamics such as the loudness, melodies and harmonies, however for this research, we first begin with the temporal structures (i.e. the rhythm) of the music and the loudness and leave the other features for future work.

Typical music relating to the western world, normally have time-signatures of 3/4 and 4/4. The first count of a time-signature (i.e. the “1” in the count of “1-2-3-4” of a 4/4 time-signature or in a “1-2-3” of a 3/4 time-structure) is referred to as the *downbeat*, and in dance choreography and music conducting for example, it is usually used to indicate when a particular dance movement or instrument in the orchestra should come in to the music. It is a very important beat to detect, naturally detected by even babies [18]. In this paper, we have detected the downbeat as simply the beat with the greatest intensity (loudness).

B. Dancing Robots

Much of today’s dancing robots are created for entertainment and social interaction purposes [6][7][17], allowing people to become the entertainers or be entertained [15] themselves. Many dancing robots today are programmed to dance to specific songs or are already preprogrammed with basic movements that appear to move rhythmically to the music [1][7]. Part of the problem with this is that the repetition becomes predictable and boring. Some researchers have also decided to use vision [8][9] to make robots dance or have designed applications that interact with the robots [10][11]. While these ideas are all very interesting, the robots are limited in adaptive behaviour as their movements are limited to specific music.

Music is dynamic in nature. By making robots able to adapt to the dynamic nature of music, it not only increases the interest of the onlookers, but also provides us with an interesting problem of the cognition, perception and the psychology of humans and human behaviour to music and dance.

All these dancing robots are very different individually and dance in their own computed ways, but most share common traits; they do not *create* their own dance moves. Most are pre-programmed to do the same movements over and over again with slight modifications to movements, for example, faster speeds and amplitudes of movements or use imitation techniques, which don’t encourage the robots to *create* their own movements. This is a problem because if we are to use dancing robots as a means of entertainment and to assist in studies to do with social and human-machine interaction, interest will soon be lost as the robot movements become boring and predictable. What are needed are more creative and dynamic movements for dancing robots that can learn and adapt to human preferences so as to keep their interest. There has been little progress in this direction

although there are some that are working close in this direction [1][20]. This research introduces a step in this direction.

C. Dancing Through Trial-and-Error

Reinforcement Learning (RL) [16] is a learning framework inspired from psychology, where agents learn and interact with the environment using a trial-and-error approach. RL typically consists of an agent in an environment and a learning problem divided up into states s_t , actions a_t and rewards r_{t+1} . The agent has a goal and the reward scheme designed should help the agent achieve that goal.

We use the Sarsa algorithm from RL as the main algorithm for this work. The Sarsa algorithm is an *on-policy* algorithm, meaning it has the ability to continuously change the values of state-action pairs while learning, and has the ability to consider sub-optimal actions to perform. The idea is to learn the value of actions in states (typically denoted as $Q(s_t, a_t)$) and estimate policies (i.e. the mapping of actions to states) [16]. It is this sequence of events that makes the robot learn and interact with the environment. It is this algorithm that we adopt in this paper as part of the learning framework, since according to psychology and animal behaviour studies, the Sarsa algorithm is the closest to the way how humans and animal behave [14].

The traditional method of the Sarsa algorithm is to receive an immediate reward after each transition, and while this may ensure faster learning, it is not a practical learning method for dancing robots as it is not how humans generally learn, except perhaps if the rewards are internal, for example, a measure of satisfaction and pleasure. However, we show later how it can be used in dancing.

It is possible that rewards can be delayed in which case, the problem now is how best to distribute the rewards over many transitions. To satisfy the delayed rewards problem in dancing robots, we use the traditional Sarsa(λ) algorithm where “ λ refers to use of an eligibility trace” [16] (denoted by $e_t(s, a)$) - a short-term memory parameter applied to each state-action pair and it resembles, which state-action pairs are eligible for learning. This was applied to experiments 2 and 3 described in the next section.

One of the main problems that reinforcement learning systems face is balancing what it already knows (exploitation) and trying out new actions (exploration). There are many different solutions to this and we do not make it our concern in this paper. We use the traditional Softmax action-selection method to gradually explore and exploit actions because, out of all the other traditional RL action-selection methods, the Softmax algorithm is the one most shown to be closer to humans and animal behaviours [4] compared to the other traditional RL action-selection

methods. We used the Softmax algorithm in the experiments to select the actions it has knowledge of, as well as to *create* new actions (shown in Experiment 3 in the next section).

Like many real life problems, there is often more than one goal to achieve and furthermore, goals can change. Dancing is no exception. For people first learning to dance or learning a new dance, one of the main things to learn is to get the timing right in response to the rhythm of the music being played. This sometimes requires many attempts to get this right. In otherwords a trial-and-error approach is adopted. It therefore seemed natural to integrate the RL framework to dancing since both follow the same underlining principle of trial-and-error.

III. IMPLEMENTATION

The idea of applying RL to learning to dance to the rhythm was divided up into three experiments. The first experiment was to make the robot dance to the beat; the second experiment was where the robot extracted the downbeat in real-time and danced *creatively* to it, and the third experiment was where the robot demonstrated the recognition of the loudness in music. We show how we can use the Sarsa and Sarsa(λ) algorithms from traditional reinforcement learning, and modify the parameters of this algorithm to achieve the different ways in which robots can dance.

With reference to this research, the external environment was the music and the feedback from the trainer; the states were where the algorithm was in relation to the music, and the agent's actions were any predefined decisions the algorithm could make e.g. what actions to move at different parts of the music or when to do actions.

To extract the beats in real-time, we used a simple beat extraction algorithm [3], which works on the idea that monotonous sound, by definition, has the same continuous energy (i.e. loudness) and so therefore, sounds the same and a beat can only be heard if the energy of the signal is more than the previously perceived sound. The software we used to demonstrate the robot dancing was the Webots [19] software package from Cybernetics. All work was integrated into one application and written all in C++.

All experiments were made to store the history and re-run again to evaluate what the robot had learnt. A complete run or termination criterion is referred to in this paper as a *trial*. Each time a music signal was played, it marked the beginning of a new trial and the robot would continue learning, based on what it had learnt previously. Below are the descriptions of each of the experiments:

Experiment 1 (Dance To The Beat): In the first experiment, we programmed a virtual robot to learn to dance to the beat and tested it using selected pieces of music. The main music file used was “faded pictures” by Case & Joe. Figure 2 shows an illustration of the idea. The robot could

effectively perform only one action any time before or after the beat, which in turn would produce immediate rewards. Figure 1 shows the Sarsa algorithm applied to this experiment.

1. Initialize $Q(s_t, a_t)$ to 0.
2. Wait till beat begins.
3. While the beat is being played:
  - a. Initialize s_t
  - b. Randomly decide when to do a_{t+1} (i.e. 'on-the-beat' or 'off-the-beat')
  - c. Check if decision for a_{t+1} in s_t is in knowledge base:
    - i. If it doesn't exist, simply "select" this decision for a_{t+1} .
    - ii. If it does exist, then generate a random number and check if the softmax of this decision is larger or not than the random number:
      1. Initialize offset parameter
      2. If random > offset && random < softmax: select decision for a_{t+1} .
      3. Otherwise check if random > offset && random < offset + softmax. If condition is met, select that decision for a_{t+1} . Otherwise drop this decision and randomly decide on another decision
      4. If no decision is suitable in the complete knowledge base, go back to line 'a.'
  - d. Do a_{t+1} and check if it is done 'on-the-beat' or 'off-the-beat'
  - e. Set s_{t+1} (i.e. 'on-the-beat' or 'off-the-beat')
  - f. If this is the first action then set s_t to 0, otherwise set to s_{t-1} (i.e. previous decision)
  - g. Update Q values:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

$$s_t \leftarrow s_{t+1}; a_t \leftarrow a_{t+1};$$
  - h. Update Softmax
4. When music stops, dancing stops.

Fig. 1: Traditional Sarsa algorithm applied to dancing on the beat.

Acceptable decisions of when to perform the action was any decision that encouraged the action to be within a certain predefined range, which marked the states for this experiment. The states were predefined as *on-the-beat* and *off-the-beat* and the typical actions that were available to the robot to choose from were *on-beat* and *off-beat*. Upon this selection, the robot would then perform the movement of a predefined joint (i.e. its head) to the music, in the state the robot was in.

The idea was to make the robot learn to do the single movement either in range (i.e. *on-the-beat*) or out of range (*off-the-beat*) and when a threshold had been met, it only selected the decision that gave it the highest reward.

Fig. 2: Illustration of beats for Experiment 1. t is the time for each beat occurrence and ηt is the number of seconds between each beat.

The robot was rewarded depending on when the action took place i.e. on the beat (*on-the-beat*) or anytime between beats (*off-the-beat*). There were a total number of 7 trials and each trial consisted of 100 actions, meaning that the robot had seven attempts to dance to the same song.

Experiment 2 (Dance To The Downbeat): In the second experiment, we introduced the downbeat to the robot, where it had to extract the downbeat in real-time, and apply varying predefined movements to it in order to demonstrate that it could behave in a rhythmic way to it.

We determined the downbeat by first determining the strong and weak beats of the musical signal by isolating the energies of each beat detected [3]. This was determined by setting a threshold, whereby if the energy was greater than the threshold, then a strong beat was detected, otherwise it was a weak beat. The assumption made here was that if the human ear detects beats based on the different loud sounds of the musical signal at varying periodic intervals, and the beats can be classified as either strong or weak, the loudest sound of a strong beat is likely to be a good indication of the beginning of a time-signature, i.e. the downbeat. Of course, there are more complex and efficient algorithms to detect the beat and determine the downbeat of the music, and even though this approach is not perfect or accurate enough, it worked really well for the music files we experimented with in this work.

We used a music file of simple beats as the main music file to test out the downbeat to demonstrate this idea, and programmed the robot to receive delayed rewards using the Sarsa(λ) algorithm. The robot was first rewarded for doing the actions on the correct beats, but then based on the modification of the internal learning coefficients, i.e. the learning rate (α) and the discount rate (γ), different actions were applied to the downbeat. The predefined actions were of the following formation:

Action1 = “head, A-B-A, 90 degrees upwards”

Action2 = “leg1 and leg2, 45 degrees sideways”, A-B-A

Action3 = “leg1, A-B-A, 90 degrees upwards”

Action4 = “leg2, A-B-A, 90 degrees upwards”

Action5 = “leg1 and leg2, A-B-A, 90 degrees upwards”

Where A is the initial home position of the joints and B is the moved position of the joint. Together, both A and B formed two movements into one motion (action). This meant that the robot was always in the same state i.e. position A (of joint positions), ready to do the next action.

Initially, the robot randomly selected actions on any beat, but then would gradually learn to only do Action2 (the desired action) on the downbeat and all other actions on the other beats based on the reward scheme. Then the robot would apply this same concept, but using other actions (not Action2), for example, it may decide to do Action5 on the downbeat and only do Action4 on every other beat. This was achieved by modifying the parameters of the learning rate and the discount factor, based on the reward given, i.e. for every reward given; the learning rate and discount factor were modified accordingly to the respective actions

performed.

We programmed the robot to always select the highest Softmax value when choosing actions and a reward of 1 was always given to the desired actions performed in the right place (and 0 otherwise). The learning rate was made small initially and the discount rate was made high. However, if a reward of 1 was given to any other action (other than Action2), the learning rate was made high and discount rate low. This meant that the value of some actions gradual increased and decreased, ensuring that actions in general were only exploited (on both the downbeat and the other beats) for a certain amount of time, before trying out another action to the downbeat. This was to give the robot some sense of rhythm, but also to demonstrate how to rhythmically dance to the downbeat. Any number of actions (approx. 480 actions) was allowed to be performed per trial.

Experiment 3 (Dance To The Beat Loudness): The first two experiments all involved predefined actions and given that most dancing robots today come with predefined actions, we decided the virtual robot should be able to *create* its own dance moves in this experiment. Here, we defined the joints (and the formation) for the robot to move, and the goal was for the robot to *create* its own actions by randomly combining these joint movements to form whole actions. If the action was in the robot’s knowledge base, then the robot would select the action with the highest Softmax instead, otherwise it was a new action (i.e. no experience of it) and the robot would perform that action anyway. The robot could perform these actions on any beat; however, what was more important was the speed at which they could be moved based on the loudness of the beats detected. The speeds were set to fixed values so that we could clearly determine if the robot could actually respond to the dynamics of music.

This experiment used the same idea as described in Experiment 2 whereby the threshold for the loudness was the average of the loudest beat, however, a beat could be in one of two states – *above* the average or *below and equal* the loudness. We used a music file consisting of a simple beat pattern of a 4/4 time-signature and one joint movement to demonstrate this. The experiment was conducted with a total of 100 trials and the robot could perform any number of actions per trial (approx. 450 actions).

IV. RESULTS

All results were as expected. However in this paper, we can only show the theoretical results obtained. For the more visual results, refer to [5]. In the first experiment, the robot began dancing by randomly moving its head *on-the-beat* and *off-the-beat* and successfully balanced the exploration and exploitation of its decisions. This was continued (as shown in Figure 3) until the threshold criterion was met and only the one decision that gave it the highest reward i.e. dancing *on-the-beat*, was exploited.

Fig. 3: Experiment 1 results of the robot dancing to the beat.

Figure 4 shows the robots random behaviour in the initial stage and then its continued exploitation of the movement on the beat.

Fig. 4: Experiment 1 results of dancing *on-the-beat* or *off-the-beat*.

For Experiment 2 (Figure 5), it was only possible to show the robots performance in the initial stage (the first trial), as it was only in this stage that the robot could show evidence of responding correctly to the downbeat, based on our preference, by performing the single action (Action2) to the downbeat, and other actions on the other beats. The virtual robot demonstrated that not only was it able to detect the downbeat in real-time while dancing, but it could dance correctly to it by arranging its actions so that other actions danced in the same way to the downbeat.

Fig. 5: Experiment 2 results of the desired action learnt in the first trial.

As can be seen in Figure 5, the robot began with an average reward of 1. This was because in this particular run of the learning stage, the robot first began its dancing with the desired action (Action2). Of course, it did not know at the time that it was the desired action and so continued exploring all other actions to find out.

Similarly, for Experiment 3, the robot had quickly

discovered the desired decision (Figure 6). It had reached the optimum reward (of 1) by the first attempt in the second trial, although what was more interesting from Experiment 3 was that, even the new actions that were created were performed at the correct speed according to the beat strength (loudness).

Fig. 6: Experiment 3 results and when the new actions were created.

We used a simple beat pattern of a 4/4 time-signature as the musical signal for experiments 2 and 3. This was because, it was easier to observe the robots movements. We observed that using a real music file for Experiments 2 and 3 made it harder to judge the robots movements mainly because it was detecting all the different beats and not a single beat. Dancing of that nature is not realistic or appealing and if implemented in a real robot (which we plan to do as future work), would cause damage to the robots joints. In Experiment 1, there was only one movement, which was of the head joint and so it was possible to use real music to assess the robots performance. All the results and robot performances can be seen in action at [5].

V. DISCUSSION & FUTURE WORK

The beat is one of the first things that are needed in order to dance to music, and for many of us too familiar with western music, without it, dancing becomes much harder. When humans dance to the beat, they select a central beat to dance to and then follow that beat for some time before moving onto the next beat, whilst at the same time, keeping in time with the rhythmic pattern, so that they know when to come in and perform their next move. But there is more to dancing than just the beats.

The art of dance is influenced by other musical features such as the quality of the sound (i.e. the timbre), and the melodies and harmonies within the music piece. Then there is the mechanics of dance in itself such as the repetition of movements, dance formations and contours and our perception of dance. Dancing robots need to be aware of all these things to ensure that they can adaptively respond to the music as the music changes.

The three experiments carried out this paper can each be extended further in their own ways, such as making the robot learn to dance to the beat using continuous space and dynamic rewards as opposed to two states like we did in this paper (Experiment 1), or making the robot create specific actions based on the previous actions created (Experiment

3), or making the robot create its own actions while dancing to the downbeat (Experiment 2). However as future work, we look towards how dancing robots can respond intuitively to the dynamics of music as part of this initial direction of making robots learn how to dance.

VI. CONCLUSION

In this paper, we have shown how we can begin to use reinforcement learning in the problem of making robots dance to some of the fundamental features of rhythm, using the Sarsa and Sarsa(λ) algorithms to consider immediate and delayed rewards respectively, and the Softmax algorithm for the exploitation and exploration of actions. The results showed the correct movements were performed to the beat. We consider as future work, two main problems of dancing robots - 1) how to actually follow the music and 2) how to appear to be following the music. The former problem relates to the fundamental features of the musical signal to be extracted, and the later problem is a matter of human perception, which requires human opinions. Our aim in this research is to make dancing robots adapt their movements to different music and as the music changes.

REFERENCES

- [1]. Aucouturier, J., Ogai, Y. and Ikegami, T. (2007) "Making a Robot Dance to Music Using Chaotic Itinerary in a Network of FitzHugh-Nagumo Neurons", *ICONIP* (2) 647-656.
- [2]. Chen, Y., Repp, B., and Patel, A. (2002) "Spectral decomposition of variability in synchronization and continuation tapping: comparisons between auditory and visual pacing and feedback conditions", *Human Movement Science*, 21, 515-532.
- [3]. GameDev.Net (2009) "Beat Detection Algorithms", Retrieved June 30, 2009, from <http://www.gamedev.net/reference/programming/features/beatdetection/>
- [4]. Gray, W. D. (2007) "Integrated models of cognition systems", *Oxford University Press*, pg. 168.
- [5]. <http://www.staff.lboro.ac.uk/~coit2/>
- [6]. Michalowski, P. M. and Kozima, H. (2007 b) "Methodological Issues in Facilitating Rhythmic Play with Robots", *16th IEEE International Conference on Robot & Human Interactive Communication, (RO-MAN)*, IEEE Press, 2007, pp. 95-100.
- [7]. Michalowski, P. M., Sabanovic, S., and Kozima, H. (2007 a) "A Dancing Robot for Rhythmic Social Interaction", *16th IEEE International Conference on Robot & Human Interactive Communication, Jeju, Korea*.
- [8]. Nakaoka, S., Nakazawa, A., Kanehiro, F., Kaneko, K., Morisawa, M., Hirukawa, H. and Ikeuchi, K. (2007) "Learning from Observation Paradigm: Leg Task Models for Enabling a Biped Humanoid Robot to Imitate Human Dance", *Int'l J. Robotics Research*, vol. 26, no. 8, pp. 829-844.
- [9]. Nakazawa, A., Nakaoka, S., Ikeuchi, K. and Yokoi, K. (2002) "Imitating Human Dance Motions through Motion Structure Analysis", *IROS*, pp. 2539-2544.
- [10]. Oliveira, J., Gouyon, F. and Reis, L. P. (2008) "Towards on Interactive Framework for Robot Dancing Applications", *Proceeding of Artech 2008 - 4th International Conference on Digital Arts Universidade Católica, Porto, Portugal, November 7-8*.
- [11]. Oliveira, J., Gouyon, F., and Reis L. (2008) "Robot Dance based on Online Automatic Rhythmic Perception", *3rd International Workshop on Intelligent Robotics*
- [12]. Patel, A., Iversen, J., Chen, Y. and Repp, B. H. (2005) "The influence of metricality and modality on synchronization with a beat", *Experimental Brain Research*, 163, 226-238.
- [13]. Repp, B. (2005a) "Rate limits of on-beat and off-beat tapping with simple auditory rhythms: qualitative observations", *Music Perception*, 22(3), 479-496.
- [14]. Scholarpedia, <http://www.scholarpedia.org/article/SARSA>, accessed 12/07/08
- [15]. Shinozaki, K., Iwatani, A. and Nakatsu, R. (2007) "Concept and Construction of a Robot Dance System", *IJVR* 6(3): 29-34.
- [16]. Sutton, R. S. and Barto, A. G. (1998) "Reinforcement Learning: An Introduction", *MIT Press, Cambridge, MA*.
- [17]. Tanaka, F., Fortenberry, B., Aisaka, K., and Movellan, J. (2005) "Plans for Developing Real-time Dance Interaction between QRIO and Toddlers in a Classroom Environment", *Proceedings of 2005 4th IEEE International Conference on Development and Learning (ICDL)*, pp. 142-147, Osaka, Japan.
- [18]. Universiteit van Amsterdam (2009, January 27) "Newborn Infants Detect The Beat In Music", *ScienceDaily*. Retrieved August 23, 2009, from <http://www.sciencedaily.com/releases/2009/01/090127083540.htm>
- [19]. Webots. <http://www.cyberbotics.com>. *Commercial Mobile Robot Simulation Software*. (Accessed 20/07/2009)
- [20]. Weinberg, G., Driscoll, S., and Parry, M. (2005) "Musical Interactions with a Perceptual Robotic Percussionist", *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)* Nashville, TN.