Dual Flexible 7 DoF Arm Robot Learns like a Child to Dance using Q-Learning

Sulabh Kumra
Electrical and Microelectronic Engineering
Rochester Institute of Technology
Rochester, NY, USA
sk2881@rit.edu

Ferat Sahin
Electrical and Microelectronic Engineering
Rochester Institute of Technology
Rochester, NY, USA
feseee@rit.edu

Abstract - *Many attempts have been made by researchers and scholars to make people feel more conversant to robots. One such example is the dance performance of an Entertainment Robot. In most cases, the challenge to program dance motions for a robot and synchronize them has been too heavy. In addition, pre-programmed dance moves and synchronization information are useful only for a specific music track and are useless for any other. To solve these problems, we developed a new system that can make a robot learn dance moves according to the input music track. The system comprises of two main parts: the first is a beat extraction system for music track; and the second one is a system that learns dance motion for Baxter. In the first part, music track is analyzed using STFT and peak-to-peak time duration is computed. This gives the beats per minute (BPM) of the given music track. The second part takes the BPM and duration of track and feeds it to the developed Q-learning algorithm to make Baxter learn dance moves and synchronize dance motion to beat rate.*

Keywords: Q Learning, Learn dance, beat extraction, Robot Dance, Baxter.

1 Introduction

Dance has always played an important role in human development and behavior. From a complex point of view, dance can be seen as a mean of nonverbal communication, although people are usually interested only in its artistic and playful side. Most of the times dance is strongly associated with music and its movements depend on well-defined music properties. Rhythm is the music element that most influences the dancing performance, and it includes several aspects such as the beat, the meter and the tempo. Although dance is something commonly done by humans and some animals, it seems appropriate to extend it to robotics. This extension will for sure provide new form of entertainment.

From a researcher's view point, robot dance is one way to show the developments of robotic technology. Especially, a robot's synchronized motions with certain music can make some people think that the robot is intelligent as it seems to understand the music. But most of the robots are pre-programmed for a specific music. All the movements are made manually and the synchronization process is also done manually. This process has one advantage that the robot's dance is at a choreography level but it also has a disadvantage that programmers have to decide what kind of motions will be shown and when those motions should be shown. Even if they accomplish these tasks, the process is highly inflexible and can't be applied to delayed music input or any other (not pre-programmed) music input. Other methods employed involve direct interaction with, or imitation of, human participant.

For these reasons, we thought that two disadvantages of current robot dance technology could be solved by giving the robot's ability to dance with simple rhythmic motions just like those of humans. Thus, we proposed a system which can recognize the beats of the music input and then the robotic system learns on its own to decide which motions should be selected for which beat. This way, the robot will learn to dance very much like how a human child learns a task on its own by trying all possibilities of a task.

In this paper, we first discuss related works in section II. Then, in section III and IV, we introduce the system configuration of the proposed robotic system, which consists of a beat tracking system and a robot motion learning system. In section V, we examine the experiments performed, to check the validity of the proposed learning system. We also discuss the limitations and future work.

2 Literature Survey

As the public has begun to have more interest in the robotic field, we can easily see that many companies and researchers are trying to prepare and provide events with robotics system. For example, 20 Nao Robots, Aldebaran Robotics' humanoid robot, demonstrated a synchronized dance on France Pavilion Day [1]. Because a dancing robot can be an important part in such events, there have been many related studies for dancing robots. There are several different approaches that may be used in the generation of dance movements in humanoid robots [2] [3] [4]. Worldwide, robotics and artificial intelligence researchers are making attempts to make robots dance to the sound of music tracks [5], and make them participate in collaborative musical performances with humans [6]. One method that was used before was in which different joints positions of the robot are set by the user for different time

978-1-4799-7611-9/15/$31.00 ©2015 IEEE

intervals and using a smoothing algorithm these frames are joint together and a dance motion is generated. Another approach and a more popular one is to make the robot dance through imitation learning in which the robot observes the dance motions of the human and extracts these motions and transfers them to their robotic joints [7], [8]. The method being used in the paper is based on non-IEC approach in which a fitness function is developed and *keyframe* concept is also used in which values will be chosen by a genetic algorithm [16], [17].

Several methods have been used till date for making a robot dance as mentioned before but making it to learn dance with both IEC and non-IEC methods and comparing both methods and making it to learn dance by synchronization with the beats of the music has not been implemented. The first advantage of the proposed algorithms is to try and reduce human inputs and make the robot learn dance-like behaviors using less or no human interactions. Secondly using reinforcement learning we make the robot dance in sync with the music and improve its dancing to music with this approach. Three diverse Q-learning algorithms have been used to learn complex behaviors using layered Reinforcement Learning [9]. In [10], as a preliminary, they developed a general experience replay framework which can be combined with essentially any incremental reinforcement learning technique and instantiate this framework for approximate Q-learning and SARSA algorithms. In another paper, we can see an adaptation mechanism based on reinforcement learning that can read subconscious body signals from a human companion, and can then use this information to adjust gaze meeting, interaction distances and motion speed and timing in human-robot interactions [11]. Reinforcement Learning is used with Decision Trees, where it uses decision trees to learn the model by generalizing the relative effect of actions across the states [12]. In this, the combination of the learning approach with the targeted exploration policy enables fast learning of the model. Kamio. S et al integrated the technique of genetic programming and reinforcement learning to enable a real robot to adapt its actions to the real world environment [13].

Marek P. Michalowski et al. have pursued research in the same manner [14]. They developed a robotic system that can interact with users. They used a small robot, Keepon, which is able to synchronize its movement with those of a child on a pressure-sensor board. The child on the board dances to the given music and the pressure-sensor sends data to the robotic system. Then, Keepon can synchronize its movements with those of the child's rhythmic dancing. With the system, the researchers showed the importance of rhythmic synchrony in social interaction. Guy Hoffman and Keinan Vanunu [15] performed research in a similar way as well. They studied various effects of robotic companionship on music enjoyment and agent perception.

3 System Configuration

The system consists mainly of two systems; the beat tracking system and the robot motion learning system. Considering the musical input of the system, how and how deeply to analyze the music is one important point to consider. For this, we divided the types of dance into two. The first one dances with simple rhythmic motions, like arm and hand movements; the second one requires the dancer's own interpretation of the music and own know-how to express their interpretation. As can be guessed, anyone can dance to the first type of music but only experts can dance to the second type. Since we are using Baxter to perform the dance moves, it can only perform arm movements, we concluded that we should aim for the first type of dance. In the first type of dance, humans don't think much and just track the basic beat. Therefore, we concluded that the most important feature is the basic beat and that the effects of the other features are not significant.

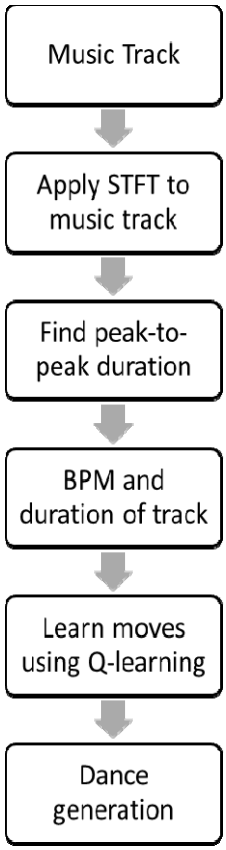

Figure 1: Overall configuration of system

The overall system configuration is shown in Fig.1. This system consists of two parts which includes real-time beat extraction music system and a dance system to make robot learn dance moves. The first system takes a music track in .mp3 or .wav format and applies Short-term Fourier Transform (STFT) and finds the beats per minute (BPM) using the peak to peak duration. This is then fed to the robotic system, where the robot learns which move is

possible for which beat and creates a Q matrix, consisting the probabilities (Q values) for going from one pose to another. These values are then used to perform the dance, i.e. complete sequence of movements. The two systems were developed on ROS based platform and coded in Python.

4 Processing Music Input

The method that is explained in this part will extract the beat rate of real-time music. Audio beat tracking algorithms commonly begin with a transformation of the input signal into an intermediary signal, often referred to as onset detection function, between the audio data and output beats. For onset detection function, we adopt a more utilitarian method, proposed by Duxbury [16], which takes account of both energy and phase information of the signal.

At first, we calculate the STFT of a mono musical audio signal with f_s = 44100 kHz sampling rate, which is an absolutely indispensable time-frequency analysis in traditional beat-tracking algorithm. Given spectrum information, onset detection function Γ(m) can be computed. For a full derivation see [16]. Fig. 2(b) shows an example of the amplitude spectrum of STFT while Fig. 2(c) shows the onset detection function of a piece of strongly rhythmic rock music. Real-time beat tracking aims at extracting beat period (the interval between successive beats) and beat alignment (the offset from the beginning of the frame to the first predicted beat within this frame) in order to locate beats within next DF frame. Inspired by Davies' research [17] on two-state switching model, we apply the general state model to our real-time module. The main idea of this stage is to use the analysis of the immediate past 6s audio data for the prediction of beat occurrences within the upcoming 1.5s.

To infer beat period, an unbiased autocorrelation *ACF(l)* of a modified detection function is calculated using:

$$ACF(l) = \frac{\sum_{m=0}^{B_f-1} \acute{\Gamma}(m)\acute{\Gamma}(m-l)}{B_f - l} \quad (1)$$

where l = 0,1,…,B_f-l corresponds to the lag of *ACF(l)* and Ǵ(m) represents the modified Γ(m). Refer to [17] for the derivative process of Ǵ(m).

Given the lag, the music tempo, in beats per minute (BPM), can be estimated by:

$$tempo = \frac{60}{l \times \Delta t} \quad (2)$$

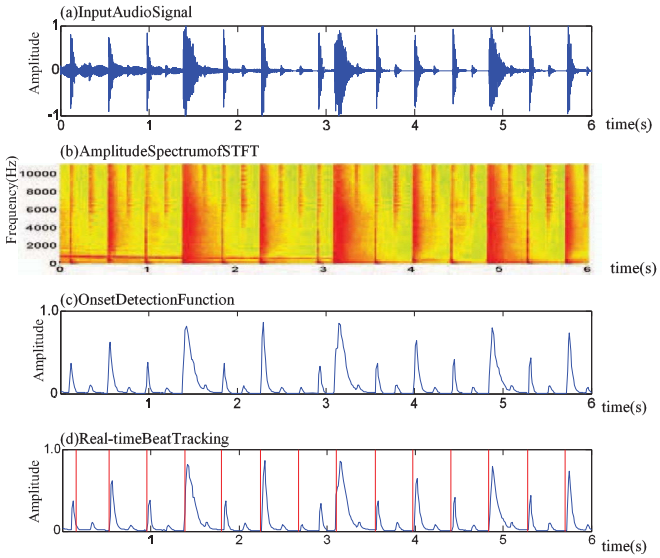

Figure 2: Typical beat tracking steps. (a) The input musical audio signal. (b) Amplitude spectrum of STFT of the signal. (c) Complex domain onset detection function. (d) A demonstration of beat tracking with dotted vertical lines representing beat locations.

Then, the autocorrelation function is passed through a shift invariant comb filter bank. Next, the beat period τ_r is identified as the index of the maximum value of the output sequence of comb filter bank. Note that $\tau_r \leq \tau_{max} = B_h$ is constrained; that means there is at least one beat for a duration of 1.5s.

The process of beat alignment induction is similar to the way in which we extract beat period. The beat alignment is identified by cross-correlating the onset detection function with an impulse train equally spaced by τ_r. Fig. 2(d) gives a demonstration of real-time beat tracking.

5 Learning Algorithm

5.1 The Algorithm

The proposed learning algorithm is Q-Learning [18]. It is a learning strategy that estimates the value functions of the status and action, which was first proposed by Watkins in 1989. The Q-Learning studies the mapping from environment status to strengthen the signal value function. In the process of interacting with the environment, the robot attempts to discover which actions can generate the most reward; the selected action not only affects the current reward and the next status, but also affects all subsequent rewards.

The Q-Learning study process is as follows: in the each time t = 0,1,2..., the robot achieving the current status $s_t \in S$ by observing the environment, S is the set of the

feasible environment status; according the current status s_t choosing an action $a_t \in A_{(St)}$, $A_{(St)}$ is the set of the feasible actions under the current status s_t ; executing the action a_t and receiving a reward value $r_{t+1} \in R$ in time t+1, then reaching the new status s_{t+1}; the target of study is learning an optimal strategy $\pi: S \times A \rightarrow [0,1]$, the strategy $A = \cup_{s \in S} A_{(s)}$ can make the total sum of the reward $V^{\pi}(S_t) = \sum \gamma^t r_t$, $0 \leq \gamma \leq 1$ is maximum or minimum, and γ is a discount factor between 0 to 1.

Defining each agent's goal is to maximize the expected sum of rewards and punishments from an arbitrary initial state by learning. The task is to learn a strategy $\pi: S \rightarrow A$, which can map the current state, *s,* to the expected behavior *a* and execute the behavior in the state, *s,* . The behavioral decision-making π is a Markova decision process which can be achieved through the interaction of the agent and the environment, and at least one optimal decision π is determined. When the environment model is unknown, agent can obtain the expected sum of rewards and punishments $Q(s,a)$ when action *a* is taken in the current state, *s,* by learning evaluation function Q: $S \times A \rightarrow R$, and then updating the value A through iteration. Q-learning rule is as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \beta \max_{a'} Q(s', a') - Q(s,a)) \quad (3)$$

where, *s* represents the current state, *r* denotes the reinforcement signal that is obtained after executing *a* of the state *s*, *s′* expresses the next state, β ($0 \leq < \beta$ 1) is the discounted factor, α ($\alpha > 0$) is the learning rate, in the learning process of Q-learning systems. Learning steps of each moment are given below.

Robot *Q-learning*(S,A,γ,α)
2: **Parameters**
3: *S* is the set of states
4: *A* is the set of actions for each state
5: α is the learning rate
6: γ is the discount
7: **Variables**
8: array *Q[S,A]*
9: previous state *s*
10: previous action *a*
11: initialize *Q[S,A]* with positive values
12: observe the current state *s*
13: **repeat**
14: select an action *a* using epsilon greedy policy
15: perform action *a*
16: observe reward *r* for this action and state *s′*
17: $Q[s,a] \leftarrow Q[s,a] + \alpha(r + \gamma max_{a'} Q[s',a'] - Q[s,a])$
18: $s \leftarrow s'$ **until** termination

where:
Q[s,a] : component of Q table (state, action)
s : state
s′ : next state
a : action
a ′ : next action
r : reward
α : learning rate
γ : discount factor

5.2 State and Action

In our case, we have described state as a pose of robot, i.e. a set of 14 joint angles for the two arms of Baxter. We have defined 21 unique pose for the robot, which act as 21 states for the algorithm. The sensors on robot gives the current joint angles, which lets algorithm know about the current state. An action is defined as the task of moving from one state or pose to another. At every state, the robot has a set of 20 possible actions, i.e. it can go to any one of the other 20 pose.

5.3 Action Selection Function

The action selection function selects the action with maximum Q-value for a given state with a probability of γ. Initially, the value of γ was set up low (as 0.1), to let the robot explore so that most of the time it will choose a random action. Though, a low value of γ will make the robot a slow learner, but it will allow the robot to explore all actions. Later, the value of γ was set high (as 0.7). For higher values of γ, each new step will dominate the previous Q-value and make the robot seem like it has no memory of the previous conditions it learnt. This was done only in the advanced robot learning process.

5.4 Update Function

The Q-values of all state-action pairs are stored in a 21×21 matrix, which was randomly initialized at the beginning. We initialized all the values as positive 10 (being optimistic). 'Q-learning algorithm require an update function that evaluates the result of an action performed in a given state in order to reward the correct behavior by a scalar' [20]. If the action is correct or suitable a positive scalar reward increases the Q-value and if the action is incorrect or unsuitable a negative scalar penalty decreases the Q-value. A set of rules dictated by the update function helps to generate certain dance move on the robot. Hence, the update function, coefficients, rewards and penalty need to be carefully chosen in order to remove the undesired behaviors and to develop with the desired behaviors. The reward or penalty are decided in our algorithm is as follows: If both the arms of the robot reach the next pose from the current pose in 4 beats, it gets a reward of +2. If only one of the arm reach the next pose from the current pose, it gets a reward of +1. In the case when both the arms don't reach the next pose in 4 beats, a negative reward of -1 is given.

6 Results & Discussion

The developed algorithm was tested on a dual flexible 7 DoF arm robotic platform called Baxter [19]. It is a new

collaborative industrial robot that is designed to work with people without the need for safety cages. It provides a robust, safe and affordable platform for innovation. It has been specifically designed with keeping in mind the needs of academic labs and corporate research departments.

Baxter was made to learn to dance on two songs with different beat rate. The beat detection algorithm extracted the BPM and length of the music track. Table 1 shows the comparison of original beat rate with the obtained beat rate for each track.

TABLE 1: Detected beat rate

Track	*Original BPM*	*Detected BPM*
We Will Rock You	83	80
Dangerous (Michael Jackson)	111.89	113

The extracted BPM and length of the music track was fed to the robotic system. Robot then tried all possible moves and received the reward based on the policy. After more than 1000 moves, we decided to stop the learning as Baxter has learned the moves according to the given music track. The learned moves are in the form of Q-values which are stored in the Q[s,a] matrix. To perform the dance, Baxter selects the next move, using the Q[s,a] matrix, which will lead to maximum reward for current and future moves. The trajectory for moving from one pose to another was produced by a ROS based package for Baxter called Joint Trajectory Action Server (JTAS). Fig. 3 demonstrates Baxter dancing on 'We Will Rock You' after learning the dance moves.

Using the same approach, Baxter was also made to learn to dance on 'Dangerous' by Michael Jackson. It was observed that Baxter was making faster moves while dancing on 'Dangerous' as compared to while it was dancing on 'We Will Rock You'. While dancing on 'Dangerous', Baxter was selecting moves which can be completed in 2.1 seconds and while dancing on 'We Will Rock You', Baxter was selecting the moves which can be completed in 3 seconds. It was also observed that this was actually proportional to the beat rate of the music tracks, which verifies correctness of the algorithm.

Figure 3: Demonstration of Baxter dancing on 'We Will Rock You' after learning the moves.

7 Conclusions

In this paper, we present two systems which makes a system that makes a robot learn to dance similar to how a child learns to dance by exploring all the possibilities. With our beat extraction system, we were able to get the beat rate of the given music track. Q-learning system was developed to make the robot learn dance moves based on the beat rate and duration of the music track. With this system of systems (SoS), we don't need to pre-program the dance moves and synchronize the robot's motion for dancing. Also, this SoS has plasticity that it can adapt to various types of music tracks. However, there is a limitation in the beat extraction algorithm. The algorithm works well only with the music tracks with constant beat rate. A system can be developed to detect in real time, the change in the beat rate and the same learning algorithm will be able to learn moves according to the continuously changing beat rate.

References

[1] “World Premiere: 20 Nao Robots Dancing in Synchronized Harmony”, http://www.youtube.com/watch?v=4t1NWH6G1f0, 2010

[2] Ju-Hwan Seo; Jeong-Yean Yang; Jaewoo Kim; Dong-Soo Kwon, "Autonomous Humanoid Robot Dance Generation System based on real-time music input," RO-MAN, 2013 IEEE, pp.204,209, 26-29 Aug. 2013

[3] Eaton, M., "An Approach to the Synthesis of Humanoid Robot Dance Using Non-interactive Evolutionary Techniques," Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on , vol., no., pp.3305,3309, 13-16 Oct. 2013

[4] Jian Sun; Jun Cheng, "A robot dance system based on real-time beat prediction," Automatic Control and Artificial Intelligence (ACAI 2012), International Conference on , vol., no., pp.287,291, 3-5 March 2012

[5] J.-J. Aucouturier et al., "Cheek to Chip: Dancing Robots and AI's Future," IEEE Intelligent Systems, vol. 23, no. 2, pp. 74-84, 2008

[6] G. Weinberg, Robotic Musicianship - Musical Interactions Between Humans and Machines. I-Tech Education and Publishing, 2007, no. September, ch. Robotic Musicianship, p. 22

[7] Maja J.Mataric, “Getting Humanoids to Move and Imitate,” Intelligent Systems and their Applications, IEEE, Volume:15, Issue:4

[8] Katsushi Ikeuchi; Takaaki Shiratori; Shunsuke Kudoh, “Robots That Learn to Dance from Observation”, IEEE Intelligent Systems

[9] Kao-Shing Hwang; Yu-Jen Chen; Chun-Ju Wu, "Fusion of Multiple Behaviors Using Layered Reinforcement Learning," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on , vol.42, no.4, pp.999,1004, July 2012

[10] Adam, S.; Busoniu, L.; Babuska, R., "Experience Replay for Real-Time Reinforcement Learning Control," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on , vol.42, no.2, pp.201,212, March 2012

[11] Mitsunaga, N.; Smith, C.; Kanda, T.; Ishiguro, H.; Hagita, N., "Adapting Robot Behavior for Human--Robot Interaction," Robotics, IEEE Transactions on , vol.24, no.4, pp.911,916, Aug. 2008

[12] Hester, T.; Quinlan, M.; Stone, P., "Generalized model learning for Reinforcement Learning on a humanoid robot," Robotics and Automation (ICRA), 2010 IEEE International Conference on , vol., no., pp.2369,2374, 3-7 May 2010

[13] Kamio, S.; Iba, H., "Adaptation technique for integrating genetic programming and reinforcement learning for real robots," Evolutionary Computation, IEEE Transactions on , vol.9, no.3, pp.318,333

[14] Marek P.Michalowski, Reid Simmons and Hideki Kozima, “Rhythmic attention in child-robot dance play”, The 18th IEEE International Symposium on Robot and Human Interactive Communication, 2009

[15] Guy Hoffman, and Keinan Vanunu, “Effects of Robotic Companionship on Music Enjoyment and AgentPerception”, 8th ACM/IEE International Conference on Human-Robot Interaction, 2013

[16] C. Duxbury, J. P. Bello, M. Davies, et al. “Complex Domain Onset Detection for Musical Signals”,Proc. Of 6th Int. Conf. on Digital Audio Effects, London, U.K., 2003.

[17] M. Davies and M. Plumbley. “Context-dependent beat tracking of musical audio”,IEEE Trans. Audio, Speech, Lang. Process., vol. 15, no. 3, pp. 1009-1020, (2007).

[18] Watkins C, Dayan P, Q-Learning, Machine Learning, 1992, 8(3):279-292.

[19] Fitzgerald, C., "Developing baxter," Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on , vol., no., pp.1,6, 22-23 April 2013.

[20] Ray, Dip N, Amit Mandal, Somajyoti Majumder, and Sumit Mukhopadhyay. "Human-like gradual learning of a Q-learning based Light exploring robot", 2010 IEEE International Conference on Robotics and Biomimetics, 2010.