2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

AMI: Adaptive Motion Imitation Algorithm Based on Deep Reinforcement Learning

Nazita Taghavi, Member IEEE*, Moath H. A. Alqatamin*, Dan O. Popa, Senior Member, IEEE*

Abstract— **In this paper, we develop a novel adaptive motion imitation algorithm (AMI) for robotic systems. Although AMI can be used in a variety of human-robot interaction scenarios, we are particularly interested in robotic rehabilitation where the robot plays the role of demonstrating and practicing challenging motion physiotherapy. During therapy, the robot first demonstrates a reference trajectory to the patient that needs to be repeated during practice and then adapts its motion to a cyclic speed and amplitude based on the patient's abilities. Using this algorithm, the robotic system learns an upper-body motion of the human user and performs a unique, similar, and easier motion based on the learned trajectory from the user. Adaptation in the AMI is based on deep reinforcement learning with deep deterministic policy gradient implemented in the Robot Operating System (ROS) environment. Experimental data collected from 11 users during upper body human-robot imitation sessions with social robot Zeno was used to show that the algorithm can learn reference elbow joint trajectories of the user in an off-line manner after just a few cycles. Finally, we also implemented the algorithm online using the Baxter robot to demonstrate its learning and playback performance.**

I. Introduction

Learning by imitation is one of the physiotherapy practices to strengthen injured muscles or learn new motion skills. Imitation is also a common practice for children suffering from autism spectrum disorders (ASD). In these individuals, it is believed that the dysfunction in the brain's mirror neuron system causes impairment in motion imitation [1] which is addressed by therapists during regular exercises. Since human therapists often get tired during imitation therapy sessions, it is considered a great idea to continue practice sessions using rehabilitation robots [2]. Past studies have shown that many patients, and especially children, elicit a positive response to these robots [3]. For example, humanoid robots have been used to learn body motions demonstrated by their human teachers [4, 5], and then perform the same motion to other impaired patients to request imitation responses. Fitter and colleagues [6] used the Nao to encourage infants with motor delay to imitate a motion of the robot such as kicking a ball, while Wijayasinghe and colleagues [7] encouraged and assessed upper body motion imitation performance of children with ASD using social robot Zeno. Zheng and co-workers [8] designed a Robot-mediated Imitation Skill Training Architecture (RISTA) for Nao to train and motivate children with ASD to learn motion skills by imitation. The robot could act as a physician, continuously demonstrate a target gesture, and sense the child imitated gesture. The robot then provided rewards or aids to the child based on their performance. In all these examples, although the robots could assess the child's performance, they only performed pre-recorded gestures and did not adapt therapy. However, numerous other algorithms have been proposed in the past to program robots to learn from demonstration from the user including the work of Tong and coworkers [9] who developed an imitation learning method based on the optimization of the dynamic movement primitives (DMPs), or studies by Martinez and Tavakoli [10] who combined learning from demonstration and robotic rehabilitation using stable estimators of dynamical systems.

One of the challenges for using a robot as a therapist to practice a physiotherapy motion, or for teaching a motion skill by imitation, is that the robot needs to adapt itself to the patient and their level of performance. Each human is unique and will exhibit different behaviors and performances during interactions with the robot. Furthermore, some subjects may be impaired physically, while others may have sensory or decision-making disabilities affecting imitation performance.

Model-free algorithms based on Reinforcement Learning (RL) can help many collaborative robotic systems learn from the user, adapt, and perform shared tasks. Many researchers have employed RL in adaptive medical and assistive devices like prostheses and exoskeletons [11]. For example, Xu and coworkers [12] developed an assistive wearable device for lower extremity rehabilitation. It consisted of a master-slave robotic system for human-robot interaction control and straightening of the injured muscles of patients during mirror therapy. RL was applied to this system to increase rehabilitation efficacy and safety. Xu and colleagues [13] proposed a shared controller based on RL to enhance the comfort level and safe operation of a walking-aid robot for elderly or disabled people. Their robot adapted to the operational control abilities of different users.

While it is a powerful technique, RL is sensitive to the so-called curse of dimensionality [14], and recently, Deep Reinforcement Learning (DRL) has been proposed to overcome this limitation. DRL can learn to map high dimensional states to values and provide continuous actions by using deep neural networks to represent policy and value functions. Rose and colleagues [15] investigated a model-free learning algorithm based on DRL to generate a user personalized desired gait pattern for a wearable exoskeleton and physiotherapy applications. The advantage of DRL in their approach was that the system could learn continuous

* Authors are with Louisville Automation & Robotics Research Institute, University of Louisville, KY 40292. (Email: {nazita.taghavi, moath.alqatamin, dan.popa}@louisville.edu)

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9812121

actions including torque values of hip, knee, and ankle actuators of the exoskeleton.

To assist patients in their physiotherapy practices with assistive devices or imitation robotic systems, two patient's challenges should be considered. Loss of appropriate motion speed is one of those main challenges in the elderly and people with mental or physical disabilities. DRL was found as a useful tool to learn a suitable speed for a specific motion of the patient and assist them to reach that speed either for motion corrections or physiotherapy purposes. For example, Sacchi and colleagues [16] proposed a DRL-based velocity control method to achieve gait symmetry in patients suffering from trans-femoral amputations. Their algorithm learned the movement from the non-amputated leg of the patient during the last gait and generated the most similar movement for an active lower limb prosthesis worn by the amputated leg to make the gait cyclic and symmetric.

Another challenge for disabled patients is to perform a motion with proper quality [17]. Usually, impaired patients are unable to fully extend their body joints during the recovery. Di Febbo [18] and coworkers used DRL for the application of functional electrical stimulation (FES) to the patient's arm muscles using a wearable elbow exoskeleton. FES is the application of the electrical charge to the injured and impaired muscles for stimulation of these muscles [19]. Their algorithm learned an optimal controller and enabled FES to assist the user to extend their elbow to reach specified elbow joint angles.

The contribution of this paper is to develop an adaptive motion imitation algorithm (AMI) based on DRL and in particular the Deep Deterministic Policy Gradient (DDPG) [20]. AMI was implemented on a robotic system to assist patients to practice target physiotherapy motions and learn from demonstration. During the interaction, the robotic system adapts the speed and shape of its robotic arm motion based on the tracked and captured sequences from the subject's arm joints angles. For physiotherapy purposes, the robot first demonstrates a default motion to the patient. Once the patient starts imitation, the system records the user hand motion and recognizes the patient's challenges regarding both speed and quality of motion. The system then adapts itself to the patient and generates an easier motion for the patient to practice. After the patient learns the easier motion, the algorithm can generate a more difficult and similar motion to the default one. As a result, this method helps the patient to practice and improve its motion's speed and quality gradually over time. A novelty of our method is that, unlike DMP, motions are parameterized for cyclic, repetitive trajectories using Fourier series coefficients. Target coefficients are then learned to best match the quality and speed of the subject using DRL after just a few demonstrated cycles. Quality of motion is then assessed using Dynamic Time Warping (DTW), which we have recently proposed as a metric to assess Human-Robot Interaction (HRI) motion quality for children with ASD [21].

For implementation and testing our algorithm, we used two robots shown in Fig.1. The first robot is Zeno, designed for children with ASD, and programmable using a laptop and a MyRIO controller running LabVIEW®. The upper arm of Zeno has only 4 degrees of freedom (DOF) and was used to demonstrate specific upper arm motions to several subjects to initiate imitation. Data collected was then used off-line to fine-tune the AMI algorithm implemented in ROS. Finally, to maintain ROS compatibility, and to utilize the full 7-DOF kinematic similarity with a human arm, we then played back upper arm imitation motions on the Baxter.

We calculated the DTW similarity cost between the user-executed motion sequences and the ones generated by the AMI algorithm. When compared to the original robot motion, results show an increase in similarity by 44%-92% for all users as a result of adaptation from our algorithm.

Figure 1. Zeno humanoid and Baxter robots in Louisville Automation & Robotics Research Institutes social robotics lab.

The paper is organized as follows: in section II we present the AMI algorithm; in section III we use data collected from human subjects to validate the algorithm offline; in section IV we implement the algorithm online on the Baxter robotic system and discuss our results. Finally, section V presents our conclusions and discusses future work.

II. Formulation of Adaptive Motion Imitation (AMI) Algorithm

The AMI HRI protocol starts with a default target motion, which in this paper is an upper-body motion performed by a healthy subject and recorded using a motion tracking system from which the upper arm joint angles are extracted. The default joints angles sequences of the arm are saved by the robotic system as a sample of a specific motion. When a new user interacts with the robot, they perform a similar motion but not necessarily with the same speed and shape or range of motion. The robotic system records the motion of the new user, calculates the Z-normalization of joint data, and learns to adapt itself to the user motion performance. In a steady state, the robot's arm with N degrees of freedom executes a periodic motion both in joint and Cartesian space imitating the movement of the human as depicted in Fig. 2.

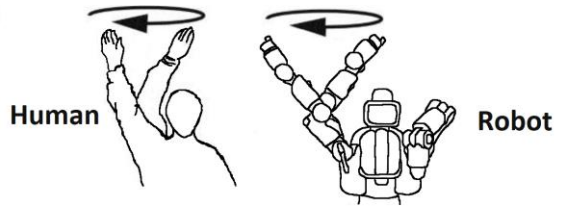

Figure 2. Depiction of motion transfer by demonstration to a robotic system.

Because demonstrated motions are periodic, each robot default joints angles sequences represented as functions of sequence indices can be approximated by a Fourier series written as:

$$F_j(i) = a_{0j} + a_{1j}\cos(w_{fj}i) + b_{1j}\sin(w_{fj}i) + a_{2j}\cos(2w_{fj}i) + b_{2j}\sin(2w_{fj}i) + \cdots, \quad (1)$$

where i is the sequence index of the joint angle time series, j is the robot joint number, $j=1..N$, w_{fj} is the frequency of the default motion, F_j is the joint angle in radians, and a_{0j}, a_{1j}, b_{1j}, …, are Fourier series constants.

To generate a similar motion with different shape and speed, we can modify the default Fourier series in equation (1) by embedding two factors for capturing differences as follows:

$$F_{mj}(i) = a_{0j} + \alpha_{ij}[a_{1j}\cos(w_{ij}i) + b_{1j}\sin(w_{ij}i) + a_{2j}\cos(2w_{ij}i) + b_2\sin(2w_{ij}i) + \cdots], \quad (2)$$

where α_{ij} is a shape factor that captures the changes in the range of the motion, w_{ij} is the frequency of motion also called the speed factor in our model, and F_{mj} is the new user joint angle sequence. For system motion adaption, the robotic system will record the motion performance of the new user and learn to select the proper shape and speed factors to generate the most similar motion sequence to the user.

In the case the robot and human upper arm kinematics are not the same, as is the case of both the Baxter and Zeno robots, we use the motion capture data, in particular, shoulder, elbow, and wrist 3D information in Cartesian space, and transform it into joint coordinates for both human and robot corresponding to robot's inverse kinematic model scaled to the size of the human arm. The kinematics of both Baxter and Zeno are well understood and can be used to compute joint angles using closed form kinematics as described in references [22] and [23]. Therefore, the joint angles extracted using motion capture according to a particular robot inverse kinematics will be played back on the same robot.

Next, we use the Deep Deterministic Policy Gradient (DDPG) algorithm to train the robotic system to imitate the target motion trajectory in joint space. We define action values as shape and speed factors in a continuous action space and F_{mj} is the system state. DDPG is an actor-critic algorithm that stores data in a buffer and during the training episodes, it randomly selects mini-batch data samples from that buffer. Deep neural networks are then used to represent the policy and action-value functions, while two other target neural networks are used to update the strategy corresponding to actors and approximate the state action-value function corresponding to the critic.

Each episode of the DDPG algorithm consists of several time steps. In our method, the number of time steps is equal to the length of the user sequence recordings from motion capture, which is usually a data set from two to three cycles of user motion. At each time-step, the algorithm selects and executes a set of action values, observes the new state, and calculates the reward. Then it stores actions, states, and rewards in the buffer. We define a reward function based on the similarity of system generated and measured subject motion sequences. The reward at each time step for each robot joint, R_{ij}, is calculated using the Euclidean distance as:

$$R_{ij} = -|(\varphi_j(i) - F_{mj}(i)|, \quad (3)$$

where φ_j is the measured sequence for joint j from the subject. The DDPG algorithm learns the new motion by maximizing the episodic rewards which are the summation of the rewards during all time-steps, I_t, of each episode:

$$R_j = \sum_{i=0}^{I_t} R_{ij} \quad (4)$$

The resulting AMI algorithm is summarized in Table 1, while the DDPG learning algorithm is shown in Table 2.

Table 1. Adaptive motion imitation algorithm (AMI)

1. Create DDPG actor, $\hat{A}(s|\theta^{\hat{A}})$, and critic, $Q(s,a|\theta^{Q})$, neural networks. s is the state, a is action, and $\theta^{\hat{A}}$ and θ^{Q} are weights.
2. Create DDPG target networks $\hat{A}^*$and Q^*
3. Initialize reply buffer, B.
4. for episode 1, M do
5. Create random noise $\mathcal{M}$.
6. Set initial state $s_{t=1,j} = \varphi_j(i=1)$, where φ_j is the measured sequence from the subject for the robot joint number, j.
7. for $t = 1$, T do
8. Select action $a_{tj} = \{\alpha_{ij}, w_{ij}\}$, add noise, $\mathcal{M}_t$, to action, execute action and observe new state, $s_{t+1,j} = F_{mj}(t+1)$.
9. Observe reward $R_{tj} = -|(\varphi_j(t+1) - F_{mj}(t+1)|$
10. Store transition in DDPG buffer, B.
11. Update critic, $\hat{A}$, actor, Q, and target networks, $\hat{A}^*$, Q^* based on the DDPG algorithm (Table 2).
12. end for
13. end for

Table 2. DDPG algorithm for updating actor and critic [20]

1. Sample a random minibatch of n transitions (s_i, a_i, R_i, s_{i+1}) from B.
2. Set $y_i = R_i + \gamma\, Q^*(s_{i+1}, \hat{A}^*(s_{i+1}|\theta^{\hat{A}^*})\,|\theta^{Q^*})$, where γ is discount factor and R is the reward.
3. Update critic by minimizing the loss function L: $L = (\frac{1}{n})\sum_i (y_i - Q(s_i, a_i, |\theta^{Q}))^2$.
4. Update the actor policy using the sampled policy gradient: $\nabla\theta^{\hat{A}} J \approx (\frac{1}{n})\sum_i \nabla_a Q(s, a\,|\theta^{Q})\,|_{s=s_i,\ a=\hat{A}(si)}\, \nabla_{\theta^{\hat{A}}}\hat{A}(s\,|\theta^{\hat{A}})\,|_{s_i}$
5. Update the target networks: $\theta^{Q^*} \leftarrow \tau\,\theta^{Q} + (1-\tau)\,\theta^{Q^*}$ $\theta^{\hat{A}^*} \leftarrow \tau\,\theta^{\hat{A}} + (1-\tau)\,\theta^{\hat{A}^*}$ where τ is the interpolation factor, $\theta^{\hat{A}^*}$ and θ^{Q^*} are target actor and critic weights respectively.

III. OFFLINE VALIDATION OF AMI WITH MOTION CAPTURE FROM HUMAN SUBJECTS DURING ROBOT IMITATION

In this section, we discuss the validation of our algorithm in designing an adaptive robot to practice upper body physical therapy exercises with the patients. In this study, the robot performs an arm hammering motion with a default speed and a range of motion for the elbow joint, as described in [21].

Our subjects were healthy adults over the age of 18 who voluntarily accepted to participate in IRB study #18.0726.

We implemented two modes of instruction for our robotic system: the adaptive and instructor modes. When the robot is in its adaptive mode, it adapts its speed and motion to the healthy user with the ability to perform a specific motion with high quality and constant shape and speed. In instructor mode, on the other hand, the robot simply demonstrates pre-selected motions to the subject, assesses their ability to follow, and gradually increases motion difficulty.

To test the adaptive mode of our system, we asked our healthy participants to mimic the motion of our robot, Zeno, and intentionally perform the hammering with constant faster or slower speeds during our experiments. To test the instructor mode of our system, we asked participants to mimic the hammering motion of the Zeno robot with the same range of motion and speed as they see from the robot. Our subjects carried a 15-pound weight during this part of the experiment to model a physical disability. We recorded the participant's hand motion using Kinect® and extracted the subject's arm joint angles as data sequences using Zeno's inverse kinematics. These data were sent as inputs to the AMI algorithm, tuned with a discount factor, $\gamma = 0.99$, and an interpolation factor, $\tau = 0.005$ in the DDPG portion.

A. Adaptive Mode of the Robotic System

For this part of the study, six subjects performed the hammering motion with approximately constant shape and constant faster speed than the default motion. We recorded the subject's motion until the Zeno robot completed three cycles of default motion, and, due to the range of motion, we selected the elbow joint data ($j = 3$) as input in the AMI algorithm. Eleven subjects completed the same experiment with a slower than default speed. Fig. 3 (a). shows the episodic rewards during 3000-episode training of the system for one of our subjects, subject S1, with the faster speed. The rewards have been maximized after approximately 500 episodes of training and converged to a maximum reward value for this subject.

Three cycles of default motion consisted of I_t (= 171) data point which is equal to the number of time-steps for training. At each time step, the algorithm selects one set of actions, $a_{tj} = \{\alpha_{ij}, w_{ij}\}$, executes them, and observes the new state and reward. In another word, the system selects I_t set of actions during each episode. Fig. 3 (b) and (c) show that the average of I_t shape factors, α_{ave}, and speed factors, w_{ave}, for each episode have converged to approximately fixed numbers equal to 1.1 and 0.31, respectively after 500 episodes when the system is trained to learn the motion of the subject S1. When the subject performs a motion with the constant shape and speed, the system finally learns to select approximately the same action values for all I_t numbers of time steps.

Fig. 4 shows the measured sequence joint angles in red from the subject S1. The default motion has been plotted in blue. After the system learns the subject's motion and calculates average shape and speed factors, it can generate a new sequence by substituting α_{ave} and w_{ave} into equation (2). If the new sequence is applied to the robotic system, the robot changes its motion speed and shape to the subject. The learned elbow joint sequence has been plotted in yellow in Fig. 4.

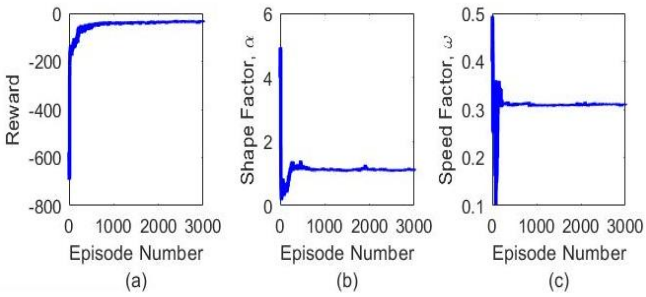

Figure 3. Training process, (a) Episodic reward, (b) Average shape factors, (c) Average speed factors.

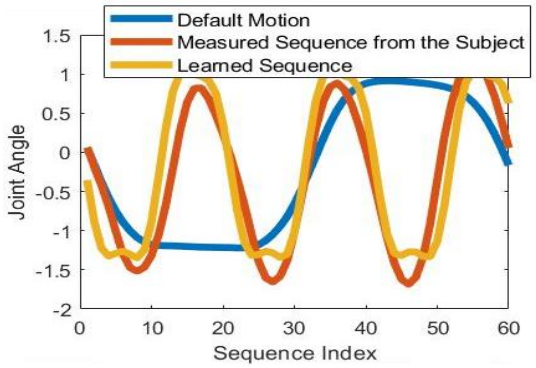

Figure 4. Generated sequence after training for the subject performing the motion with a fast speed, $w = 0.3$.

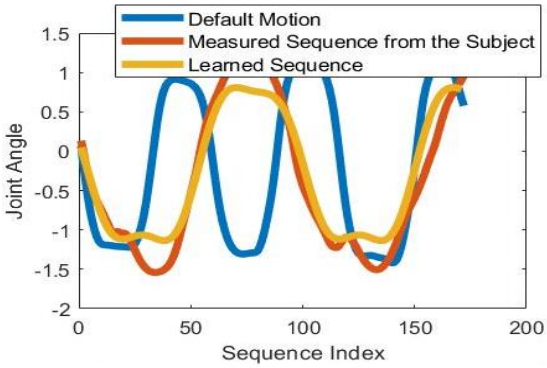

Figure 5. Generated sequence after training for the subject performing the motion with a slow speed, $w = 0.06$.

The subject's joint angles sequence, default motion, and the generated sequence after training for the subject S2, who has performed hammering slower than the default speed is shown in Fig. 5. As shown in Figs. 4 and 5, the final generated sequence matches the subject's sequence which shows that the algorithm has generated adaptive sequences. Also, the generated sequence is feasible for a robot to perform given its hardware joint limitations since the sequence is very similar to the default motion which was already executed by the robot. In our case, the sequence data should be sent to the robot joint at the rate of 22 ms to match the speed of the human subject.

After training the robotic system for eleven subjects who performed hammering with a slower speed, we calculated Dynamic Time Warping (DTW) cost between the subject and system-generated sequences with the same length of one cycle of default motion. We compared these values with the DTW cost calculated between the subject and default motion sequences for one cycle. Results shown in Table 3 reveal that the DTW cost between subject and learned sequences is significantly smaller with the average percentage decrease of

78% for 11 subjects. The smaller values of DTW cost indicate higher similarity between the subject and AMI generated sequences. Furthermore, we carried out the sequence similarity calculations for our six subject sequences who completed the experiment with a faster speed, as shown in Table 4. The average percentage decrease of DTW costs, in this case, is still significant and averages to 68% for the 6 subjects.

Table 3. DTW cost decrease percentage for subjects performing motion with slow speeds

Subject No.	1	2	3	4	5	6
% Decrease	92.4	82.6	51.5	65.2	74.4	66.3
Subject No.	7	8	9	10	11	
% Decrease	57.9	89.8	52.7	90.4	86	

Table 4. DTW cost decrease percentage for subjects performing motion with fast speeds

Subject No.	1	2	3	4	5	6
% Decrease	72.1	61.9	74.5	44.1	69.7	85.1

B. Instructor Mode of the Robotic System

The instructor mode of the algorithm is more useful for patients who want to repeatedly practice a physiotherapy exercise motion with a robot. When the patient improves their motion performance, as indicated by a low DTW cost between human and robot motion, the system generates a more difficult sequence for the patient. This method helps the patient to gradually improve their physical abilities.

To test the instructor mode of the system, we asked our participants to carry a 15-pound weight when mimicking the hammering motion of the robot with default speed. Fig. 6 shows the results of 3000-episode training for the subject S3 who carried a 15-pound weight during his motion performance. Like the adaptive mode of the system, the rewards are maximized after several episodes and average shape and speed factors converge to approximately fixed numbers although the subject was unable to control the constant speed and shape of his motion.

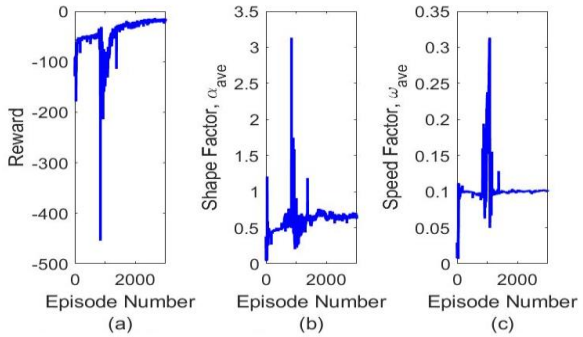

Figure 6. training results for instructor mode of the robotic system (a) episodic rewards, (b) average episodic shape factor, (c) average episodic speed factor.

We have plotted the measured sequence from the subject S3 and system states, $F_m(i)$, generated by the system after training in Fig. 7. As shown, the system learns the subject's motion and the agent's states during one episode tracks the subject's motion sequence. The subject's motion shape and speed are not constant, which means the learned α_{ij} and w_{ij} in equation (2) are different for each index of the motion, but the system finds the average shape and speed of the subject's motion by calculating average shape and speed factors. Therefore, when the system generates a new sequence using the calculated average shape and speed factors, it creates a motion sequence that is easier for the patient to practice since it is based on averaging the speed and shape of the subject's motion, and it considers unique abilities of the patient. The generated sequence has been shown in Fig. 7 for subject S3.

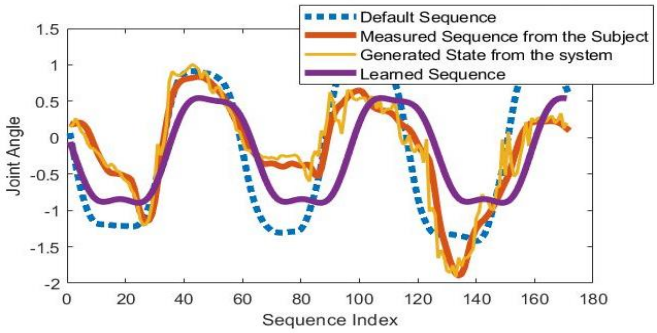

Figure 7. System generated states and final sequence for a subject carried 15-pound weight.

Seven subjects completed the hammering motion performance while they carried 15-pound weight to model physical disability. The shape and speed factors for the default motion were 1 and 0.11, respectively. We have trained our system for all the subjects and calculated average shape and speed factors. We compared these values with the default motion shape and speed factors and plotted them in Fig. 8, while default factors are plotted in red. Results demonstrate that the system selects motions with lower speed and range for most of the subjects, which are easier for them to practice.

The offline training process of AMI was done on a Lenevo, Legion laptop with an Intel Core i7 and CPU speed of 2.60 GHz with 16 GB of RAM. With these subjects, training took 2 to 5 minutes until we obtained large rewards below -10.

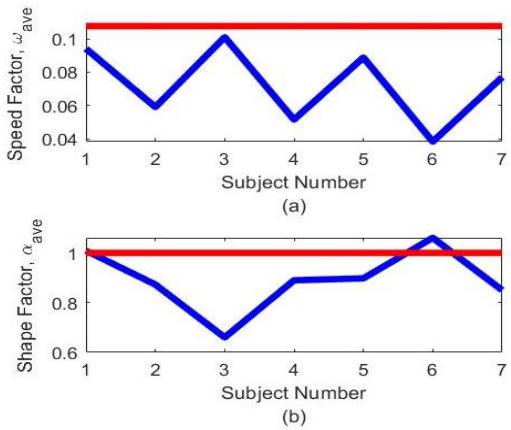

Figure 8. Average shape and speed factors for subjects carrying 15-pound weight (a) average shape factor, (b) average speed factor.

IV. ONLINE IMPLEMENTATION AND PLAYBACK OF LEARNED AMI MOTIONS

Once the AMI algorithm was validated using off-line captured data during imitation interaction with the Zeno robot, we implemented a similar experiment using the Baxter robot. Our aim was to train a motion, in this case, hand waving, to

the robot and defined a reward threshold number, R_m, at which the robot learns and starts the motion performance. For hand motion, the focus was on the elbow joint angle, which contributes the most range of motion to the human subject's upper arm. In this scenario, the AMI algorithm was implemented in Python 3 running in a virtual environment and Ubuntu 14.04. The communication with Baxter's joint motors was accomplished through a socket written in Python 2 to receive data from the AMI algorithm and communicate with the robot operating system (ROS). Specifically, we sent joint commands to the robot in its position control mode and read the joints angles from the relevant ROS topic.

We first recorded two subjects' hand waving motions, one as a default motion and the other one as a motion that must be learned by the robot. We recorded these motions using a Kinect sensor and save them as text files for the robotic system to use.

For the safety of our robot hardware, we bounded the frequency and motion ranges so if the algorithm selects actions that are out of the robot hardware range, the system will not be damaged. When the system starts training, the robot arm is in the pose ready for the hand waving. The algorithm calculates the episodic rewards and sends joint commands to the robot when the episodic reward is more than a defined reward number, R_m. We select R_m experimentally to assure that the robot starts the motion when it properly learns the trajectory. During our experiments, the machine learning system selects actions and calculates episodic rewards. Once the system generates an episodic reward higher than R_m, the joint command is calculated based on average shape and speed factors and sent to the relevant robotic joint. R_m then is updated to be the current calculated episodic reward. The robot changes the motion only when the algorithm finds a larger reward than the updated R_m. For rehabilitation applications of AMI, it is important that we define a proper initial R_m, so the robot starts the motion when it has learned the trajectory, and small changes in motion are tailored to each patient during physiotherapy practice. Also, R_m, should not be a very large number because motion training will take a very long time and make HRI impractical.

To find R_m experimentally, we first selected a small R_m = -100 for our first trial and recorded input commands to the robotic arm as well as the system output. We stopped the system when we got episodic rewards around -33. Fig. 9. shows the changes of produced sequences as inputs and system outputs over time. As shown, first the algorithm explores different actions and generates inaccurate sequences. However, for episodic rewards higher than -40, the changes intend to be minor. While Baxter performed the motion, we confirmed that for the rewards in the range of -40 to -35, human eyes cannot notice the changes in motions when the robot switched to the new motion with better rewards.

For our second trial, we selected defined a reward threshold R_m = -35. The robot performed the motion until we obtained episodic rewards around -15. Again, we confirmed that the human eye could not notice changes when the system switched the motions. Fig. 10. shows the system inputs and outputs for our second trial as well as final motion generation from the system at the reward around -15.

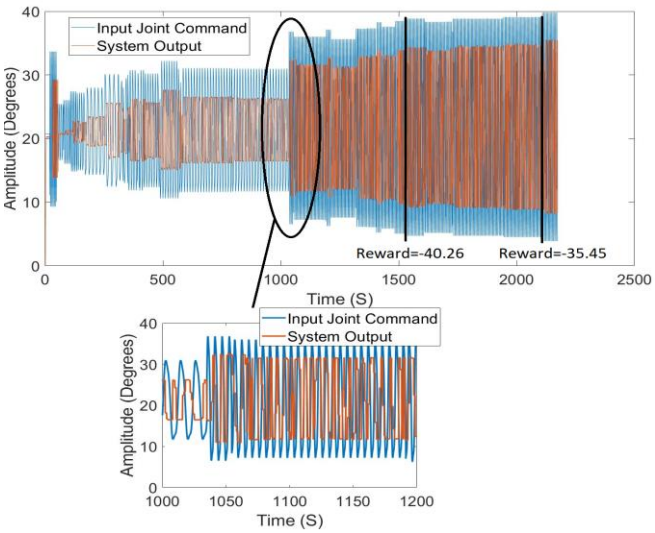

Figure 9. Imitation algorithm input to the Baxter system and output from the system during motion training for the first trial.

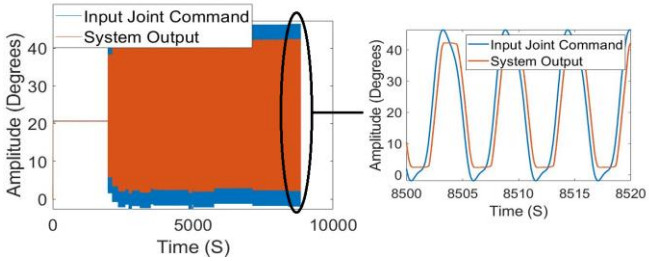

Figure 10. Imitation algorithm input to the Baxter system and output from the system during motion training for the second trial.

V. Conclusions and Future Work

In this paper, we proposed the AMI, a new adaptive motion imitation algorithm, to design an adaptive robotic therapist for upper arm imitation. The system uses the measured joint angle sequences of the patient's upper-body motions and adjusts the shape and speed of the robot's cyclic trajectories to practice an easier motion with the patient considering their unique abilities. We validated our algorithm with off-line data collected from adult human subjects and showed how the algorithm can be also implemented online for the elbow joint.

In the future, we plan to implement the algorithm on Zeno and Milo humanoid robots in conjunction with a higher-performance controller and computing hardware to decrease the training time and fidelity of motion control. We will then test the AMI algorithm with more challenging target motions, involving shoulder, elbow, and wrist joints, in support of upper arm physical rehabilitation, or for robotic interventions for children with ASD. We will address learning scalability challenges to robot arms with higher numbers of degrees of freedom (DOF) and compare the performance of AMI with other supervised learning methods.

Acknowledgment

This work was supported by the US National Science Foundation through grants SCH IIS#1838808 and EPSCoR OIA#1849213. We wish to thank Dr. Sumit K. Das for his help with the experimental results presented in this paper.

REFERENCES

[1] M. M. Y. Chan, Y.M.Y. Han, "Differential mirror neuron system (MNS) activation during action observation with and without social-emotional components in autism: a meta-analysis of neuroimaging studies." *Molecular Autism.*, vol. 11, no. 72, 2020.

[2] N. A. Malik, H. Yussof, F.A. Hanapiah, "Development of imitation learning through physical therapy using a humanoid robot" *Procedia Comput. Sci.*, vol. 42, pp. 191-197, 2014.

[3] T. Belpaeme, P. E. Baxter, R. Read, R. Wood, H. Cuayahuitl, B. Kiefer, S. Racioppa, I. Kruijff-Korbayova, G. Athanasopoulos, V. Enescu, "Multimodal child-robot interaction: Building social bonds," *J. hum. robot interact.*, vol. 1, no. 2, pp. 33–53, 2012

[4] M. Riley, A. Ude, K. Wade, and C. G. Atkeson, "Enabling Real-Time Full-Body Imitation: A Natural Way of Transferring Human Movement to Humanoids." presented at the 2003 IEEE Int Conf Robot Autom (ICRA), pp. 2368–2374. Taipei, Taiwan.

[5] J. Koenemann, F. Burget, and M. Bennewitz, "Real-Time Imitation of Human Whole-Body Motions by Humanoids." presented at the 2014 IEEE Int Conf Robot Autom (ICRA), pp. 2806–2812. Hong Kong, China.

[6] N. T. Fitter, R. Funke, J. C. Pulido, L. E. Eisenman, W. Deng, M. R. Rosales, N. S. Bradley, B. Sargent, B. A. Smith, and M. J. Mataric, "Socially Assistive Infant-Robot Interaction: Using Robots to Encourage Infant Leg-Motion Training." *IEEE Robot. Autom. Mag.*, vol. 26, no. 2, pp. 12–23, June 2019.

[7] I. B. Wijayasinghe, I. Ranatunga, N. Balakrishnan, N. Bugnariu, and D. O., Popa, "Human–Robot Gesture Analysis for Objective Assessment of Autism Spectrum Disorder." *Int. J. Soc. Robot.*, vol. 8, no. 5, pp. 695–707, November 2016.

[8] Z. Zheng, E. M. Young, A. R. Swanson, A. S. Weitlauf, Z. E. Warren, N. Sarkar, "Robot-Mediated Imitation Skill Training for Children with Autism," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 6, pp. 682-691, June 2016.

[9] X. Tong, R. Li, L. Ge, L. Zhao, K. Wang, "Imitation Learning of Human Operation Based on Visual Demonstration," presented at the 3rd International Conference on Control and Computer Vision (ICCCV), pp. 71-76, Macau, China, 2020.

[10] C. Martínez, M. Tavakoli, "Learning and robotic imitation of therapist's motion and force for post-disability rehabilitation," presented at the IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2225-2230, Banff, Canada, 2017.

[11] A. Mohebbi, "Human-Robot Interaction in Rehabilitation and Assistance: a Review," *Curr. Robot. Rep.*, vol. 1, pp. 131–144, 2020.

[12] J. Xu, L. Xu, Y. Li, G. Cheng, J. Shi, J. Liu, S. Chen, "A Multi-Channel Reinforcement Learning Framework for Robotic Mirror Therapy," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 5385-5392, Oct. 2020.

[13] W. Xu, J. Huang, Y. Wang, C. Tao, and L. Cheng, "Reinforcement learning-based shared control for walking-aid robot and its experimental verification," *Adv. Robot.*, vol. 29, no. 22, pp. 1463–1481, 2015.

[14] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," ICLR, 2016.

[15] L. Rose, M. C. F. Bazzocchi and G. Nejat, "End-to-End Deep Reinforcement Learning for Exoskeleton Control," 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 4294-4301, 2020.

[16] N. Sacchi, G. P. Incremona, A. Ferrara, "Deep Reinforcement Learning of Robotic Prosthesis for Gait Symmetry in Trans-Femoral Amputated Patients," 29th Mediterranean Conference on Control and Automation (MED), pp. 723-728, 2021.

[17] S. Sadegh Pour Aji Bishe, T. Nguyen, Y. Fang and Z. F. Lerner, "Adaptive Ankle Exoskeleton Control: Validation Across Diverse Walking Conditions," *IEEE Trans. Med. Robot. Bionics*, vol. 3, no. 3, pp. 801-812, 2021.

[18] D. Di Febbo, E. Ambrosini, M. Pirotta, E. Rojas, M. Restelli, A. Pedrocchi, S. Ferrante, "Reinforcement Learning Control of Functional Electrical Stimulation of the upper limb: a feasibility study," in Annual Conference of the International Functional Electrical Stimulation Society (IFESS), pp. 111–114, 2018.

[19] N. Taghavi, G. R. Luecke, N. D. Jeffery, "A neuro-prosthetic device for substituting sensory functions during stance phase of the gait," *Appl. Sci.*, vol. 9, no. 23, pp. 5144, Nov. 2019.

[20] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.

[21] N. Taghavi, J. M., Berdichevsky, N., Balakrishnan, K. C., Welch, D. O., Popa, "Online Dynamic Time Warping Algorithm for Human-Robot Imitation" presented at the 2021 IEEE Int Conf Robot Autom (ICRA), pp. 3843-3849. Xi'an, China.

[22] N. Torres, N. Clark, I. Ranatunga, and D. Popa, "Implementation of interactive arm playback behaviors of social robot Zeno for autism spectrum disorder therapy," in PETRA2012: The 5th International Conference on PErvasive Technologies Related to Assistive Environments, pp. 21, Heraklion Crete, Greece June 6 – 8, 2012.

[23] S. Cremer, L. Mastromoro, D. O. Popa, "On the performance of the Baxter research robot," 2016 IEEE International Symposium on Assembly and Manufacturing (ISAM), pp. 106-111, Fort Worth, USA, 2016.