



Adaptive Motion Imitation for Robot Assisted Physiotherapy Using Dynamic Time Warping and Recurrent Neural Network

Ali Ashary*

Louisville Automation Research and
Robotics Institute (LARRI), University
of Louisville, Louisville, KY 40292
USA

ali.ashary@louisville.edu

Madan M. Rayguru

Louisville Automation Research and
Robotics Institute (LARRI), University
of Louisville, Louisville, KY 40292
USA

madanmohan.rayguru@louisville.edu

Jordan Dowdy

Louisville Automation Research and
Robotics Institute (LARRI), University
of Louisville, Louisville, KY 40292
USA

jordan.dowdy@louisville.edu

Nazita Taghavi

Louisville Automation Research and
Robotics Institute (LARRI), University
of Louisville, Louisville, KY 40292
USA

ntaghavi2012@gmail.com

Dan O. Popa

Louisville Automation Research and
Robotics Institute (LARRI), University
of Louisville, Louisville, KY 40292
USA

dan.popa@louisville.edu

ABSTRACT

Robot assisted physiotherapy has the potential to reduce the workload of healthcare professionals and deliver interventions in the comfort of the home. In general, physiotherapy involves repeating a specific set of motions until an efficiency metric is reached. In robot assisted physiotherapy sessions, two major questions arise: 1) How to accurately quantify the similarity of the motion between the robot and the subject; 2) How to adapt the robot's behavior according to the subject's ability. In this paper, we address these two questions by proposing a new modular framework for adaptive motion imitation (AMI) using a deep long-short term recurrent neural network (LSTM-RNN) and segment online dynamic time warping (SODTW). Our framework uses the SODTW cost as a metric for quantifying the similarity between the motion of robot and subject. The LSTM-RNN takes the range of motion and the fundamental discrete Fourier transform (DFT) coefficients as inputs and uses them to predict a dynamic and periodic reference trajectory for the robot. By modifying the DFT coefficients based on the SODTW cost of the subject, the output of the LSTM-RNN is then adapted according to the imitation ability of subjects. The separation between the prediction and adaptation portions of our framework greatly simplifies testing, coding and improves the algorithm scalability. We tested the proposed AMI framework with 10 participants to verify its effectiveness. The results demonstrate

the validity of the proposed framework in adapting the behavior of the robot according to the subject's imitation abilities.

CCS CONCEPTS

• **Human-robot interaction;** • **Recurrent Neural Network (RNN);** • **Gesture and motion tracking;**

KEYWORDS

Robot-assisted physiotherapy, Adaptive motion imitation, Deep learning, Healthcare robotics

ACM Reference Format:

Ali Ashary, Madan M. Rayguru, Jordan Dowdy, Nazita Taghavi, and Dan O. Popa. 2024. Adaptive Motion Imitation for Robot Assisted Physiotherapy Using Dynamic Time Warping and Recurrent Neural Network. In *The Pervasive Technologies Related to Assistive Environments (PETRA) conference (PETRA '24)*, June 26–28, 2024, Crete, Greece. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3652037.3652079>

1 INTRODUCTION

Physiotherapy and rehabilitation are critical aspects of healthcare by helping patients to recover from post-surgery symptoms, injuries, and neurotic conditions [1]. In the last few decades, researchers have explored the use of many types of social and medical robots, as well as learning algorithms for robot assisted physiotherapy [2]. These contributions promise to improve patient outcomes in various ways like better movement, increased strength, and independent living [3]. Some healthcare professionals have been motivated to tailor their physiotherapy programs to involve robot enabled solutions [4]. Robot assisted physiotherapy programs have also attracted attention of medical professionals specializing in Autism Spectrum Disorders (ASD). Nao and Milo are examples of commercially available social robots with interactive functionality to assist special educators in improving the social, physical, and communication skills of autistic children [5], [2]. Research evidence to date has shown encouraging outcomes for ASD treatment programs involving social robots [6], [7]. However, issues like technology integration, training methods, additional costs, and

*This work was supported by NSF grants #1838808 and #1849213. Authors are with Louisville Automation and Robotics Research Institute (LARRI), University of Louisville, Kentucky, U.S.A., Contact: ali.ashary@louisville.edu, madanmohan.rayguru@louisville.edu

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

PETRA '24, June 26–28, 2024, Crete, Greece

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1760-4/24/06

<https://doi.org/10.1145/3652037.3652079>

standardization of treatment protocols for robot-assisted physiotherapy are still challenging for researchers and educators alike.

Imitation learning is proving to be a very effective tool in robot assisted physiotherapy for patients with ASD. Several types of imitation learning algorithms have been proposed to train robots to mimic the motions required for physiotherapy and rehabilitation sessions [8], [9]. The trained robot then demonstrates the learned motions to the patients and then records their responses for further evaluation [10]. The authors of [11] used proposed an architecture for robot assisted skill training for children with ASD. Deterministic policy gradient, approximate dynamic programming, and recurrent neural networks are some of the most used techniques for imitation learning in social robots [12], [13], [14].

Xu et. al. developed a new shared control technique based on reinforcement learning for helping subjects in walking tasks, [15]. A reinforcement learning based master slave robotics system is presented in [16] for mirror therapy of limb impaired patients. In the work [17], an adaptive ankle exoskeleton control was validated for improving the walking ability of patients. Deep reinforcement learning (DRL) approaches have the ability to solve high dimensionality problems that have long limited conventional RL techniques. Recently, in the work of Taghavi et. al. [18], we proposed a deep deterministic policy gradient (DDPG) method for adaptive motion imitation to train children with ASD. The approach predicted so called "shape" (e.g. magnitude) and "speed" (e.g. frequency) factors from the periodic recorded joint motion of subjects. However, the RL and DRL-based approaches are computationally more intensive and time consuming, as they require extensive interaction with the environment for exploring optimal policies.

Compared to RL, the RNN-based techniques excel at predicting temporal data sequences in dynamically changing tasks like physiotherapy exercises [19], [20]. The paper [21] developed an RNN-based imitation learning method for social interaction based on human-to-human interaction data. The authors of [22] proposed a gaze based imitation learning and master robot policy transfer mechanism for teaching tasks that require force sensor data. The long short-term memory RNN (LSTM-RNN) is exploited for imitation learning based on visual change based image representation [23]. A novel objective function and LSTM-RNN is utilized to predict the future behavior of pedestrians in [24]. The paper [25] proposes a new methodology for adaptive imitation learning based on the addition of dynamic constraints and parametric bias. A modified recurrent neural network (RNN) is trained by the demonstration images, the control inputs and an extra parameter called parametric bias. With this approach, the RNN was able to adapt to the variations in task execution. However, training a RNN for all the task variations is a challenging task. For example, the work [25] uses multiple dynamic constraints for training their network for different tasks.

Another challenge of the current state of the art in adaptive robot imitation [17], [22], [18], [25] is the lack of generalization and standardization of these approaches to different settings. As these works use different training strategies and network architectures, they may result in different outcomes from very similar robot assisted physiotherapy routines. If a uniform metric is used to quantify the similarity/discrepancy in the task executions of robot and subject and the same metric is utilized to train/adapt the

neural networks, then the lack of standardization can be avoided. The SODTW algorithm presented in our recent work [2] can be a suitable choice, due to invariance of signal temporal alignment. In robot assisted physiotherapy sessions, it is not only important to capture the difference in amplitude between subject and robot motion, but also the speed at which subject repeats the motion. The SODTW algorithm can capture both of these aspects in an efficient manner and therefore has the ability to standardize the evaluation of the subject's physical ability.

Contributions: To address these challenges, we propose to use a LSTM-RNN deep network for prediction of robot periodic physiotherapy motions, and the SODTW distance as a basis to adapt the robot motions based on subject responses. To accomplish this task, we reversed the input-output mapping, that was used in our earlier work on DDPG [26] to keep the input dimension constant in presence of motion variations. The major contributions of the paper are as follows:

- We propose a new modular framework of adaptive motion imitation for robot assisted physiotherapy. Unlike the earlier approaches, the reference trajectory predictor (LSTM-RNN) and adaptation algorithm (SODTW Module) can be modified independently for different types of physiotherapy sessions. This simplifies the extension of our framework and standardize the motion adaptation.
- A new training strategy is presented for multivariate LSTM-RNN, which can generate reference trajectories for the robot, based on the imitation ability of the subject. The network is trained with only the fundamental Fourier coefficients (amplitude and frequency component) and the range of periodic motion as its inputs. Once trained, the LSTM-RNN can generate a suitable reference trajectory for the Zeno robot to adapt to different variations in shape, speed, or their combination according to a fuzzy inference engine.
- The proposed LSTM-RNN + SODTW Framework was validated with the help of subject trials on our Zeno robot. In the training phase, we collected joint trajectory data from 10 subjects imitating an upper body motion demonstrated by Zeno. The data from five subjects are processed, resampled, and normalized for training the LSTM-RNN. Furthermore, we asked the other five subjects to intentionally modify the motion and used the data to test our proposed framework. The experiments show that Zeno's motion adapts well according to all the variations in motion imitation from test subjects.

The paper is organized as follows: in section 2, we discuss the proposed AMI scheme and present a new deep LSTM-RNN architecture. In section 3 we discuss the experimental results and finally section 4 presents our conclusions and discusses future work.

2 PROPOSED AMI FRAMEWORK

The proposed modular Adaptive Motion Imitation (AMI) framework is presented in Fig. 1

The framework involves a set of tasks carried out synchronously, described here. The robot-assisted physiotherapy session starts with Zeno demonstrating a specific upper arm periodic motion to

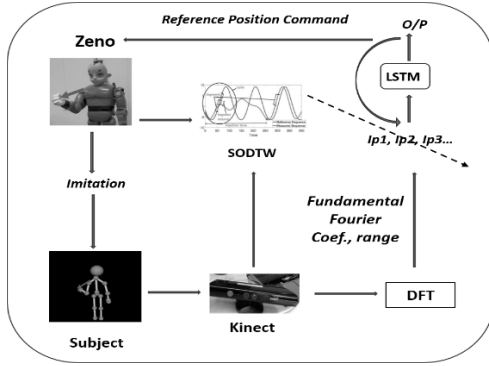


Figure 1: Our Modular Adaptive Motion Imitation (AMI) Architecture

human subjects. The subjects are asked to imitate the same motion as closely as possible. The motion of the subjects is recorded through Kinect and the joint trajectories for both human and robot are computed. The joint trajectory of the subject is then fed to a Discrete Fourier Transform (DFT) software routine to generate the first few fundamental components of amplitude (shape) and frequency (speed). At the same time, the SODTW cost is computed which measures the similarity between the recorded human motion and reference Zeno motion. The obtained fundamental DFT coefficients are then normalized to make the training uniform for all ranges of motion. These coefficients along with the range of motion are then passed through our LSTM-RNN in order to train it to predict appropriate reference joint trajectory commands for Zeno. The working of the prediction layer based on LSTM-RNN will be discussed in detail in the next section.

2.1 SODTW

The SODTW cost is used as the metric to alter the inputs (DFT coefficients) so that the trained LSTM-RNN can change the reference commands for Zeno proportionally. The reason for not using SODTW cost as another input for LSTM-RNN comes from the need to simplify the framework. In general, RNN is quite effective in mapping the temporal relationship in the input data. As SODTW cost by default is not explicitly linked to the input data, the RNN may take more computation time to learn the map for all motion variations. By keeping the SODTW cost computation as a separate entity and using it to directly change the inputs, we can achieve a similar adaptation to motion variations in speed, amplitude, or a combination of both.

The SODTW algorithm [26] is a variation of the Dynamic Time Warping (DTW) algorithm that is used to compare and align two time series sequences, typically with variations in time and speed (Fig. 2). Given a reference and a measured signal of M and N sample, the SODTW algorithm [2] calculates the difference between the samples of two signals using a dynamic programming approach:

$$D(i, j) = ||x_i - y_j|| + \min(D(i-1, j-1), D(i-1, j), D(i, j-1)) \quad (1)$$

where $i=1, \dots, M$; $j=1, \dots, N$, and $D(i, j)$ is the Euclidian norm of difference between i^{th} sample of reference trajectory and j^{th}

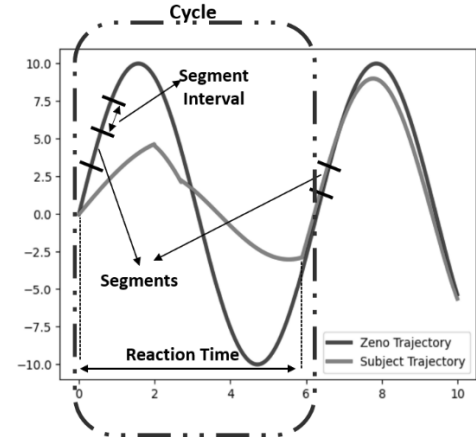


Figure 2: Segment Online Dynamic Time Warping (SODTW) scheme computes the similarity cost between robot and subject joint periodic trajectories.

sample of the measured trajectory. SODTW recursively calculates the similarity cost between two samples depending on the cost of previous samples, and $D(M, N)$ will give the final cost for entire motion sequences.

2.2 Fuzzy engine

When the subject does not follow a particular motion in a satisfactory manner of the robot, the SODTW cost becomes high. The high SODTW cost is an indication to make the physiotherapy simpler till the subject achieves good cost. Similarly, low SODTW cost points to perfect motion imitation of the subject. In such cases, the difficulty level of physiotherapy should be increased. In general, it was found that slower motions are difficult to imitate compared to faster motions (refer table 2). So, the Zeno robot's reference trajectory is made slower when the user has a lower SODTW cost and vice versa. We used this simple logic to build a rule base for a simple MAMDANI FIS (ref. Fig 3) which computes the scaling factor for the LSTM-RNN inputs.

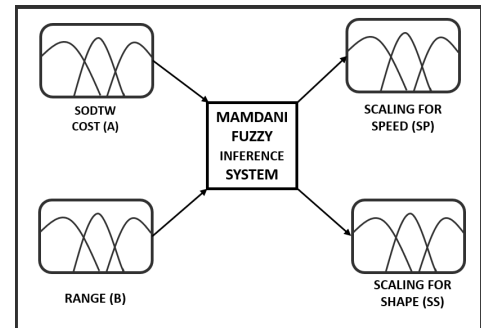


Figure 3: Fuzzy Engine to Modify LSTM-RNN Inputs

2.3 LSTM-RNN for Reference Generation

The conventional RNN architecture may suffer from gradient vanishing or explosion during the weight updates by backpropagation [24]. The LSTM architecture eradicates this issue by introducing different gate layers as well as two state variables called cell state and hidden state (see Fig. 4).

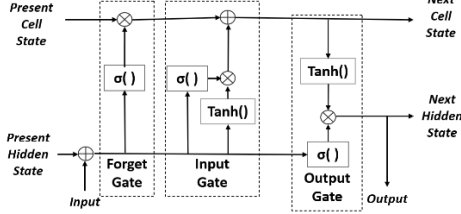


Figure 4: The Gate layout of a Single LSTM Cell

The LSTM cell exploits two well-known activation functions, σ ($\cdot$) = $\frac{1}{1+\exp(\cdot)}$ and $\tanh(\cdot)$ = $\frac{1-\exp(\cdot)}{1+\exp(\cdot)}$ to compute the intermediate variables and next cell state as well as hidden state.

To express the forward pass equations for a single LSTM cell, let's define the input vector as $x_k = [x_1(k), x_2(k), \dots, x_n(k)]$, weight matrix for forget gate as W_f , weight matrix for input gate as W_i , weight matrix for output gate as W_o and the bias vectors as b_f, b_i, b_o . Let the current cell state is c_k and the hidden state is h_k . The forward pass equations can be written as:

$$\begin{aligned} i_{k+1} &= \sigma(W_i \cdot h_k + W_i \cdot x_k + b_i), \\ f_{k+1} &= \sigma(W_f \cdot h_k + W_f \cdot x_k + b_f), \\ \tilde{c}_{k+1} &= \sigma(W_c \cdot h_k + W_c \cdot x_k + b_c), \\ c_{k+1} &= f_k c_k + i_{k+1} \tilde{c}_{k+1}, \\ o_{k+1} &= \sigma(W_o \cdot h_k + W_o \cdot x_k + b_o), \\ h_{k+1} &= o_{k+1} \tanh(c_{k+1}) \end{aligned} \quad (2)$$

where f_k, i_k, o_k and $\tilde{c}_k$ represent the current state of forget gate, input gate, output gate and intermediate cell state respectively.

The objective is to adapt the Zeno robot's motion (a periodic joint trajectory) according to the need of the subject (SODTW similarity score) and we are using a deep LSTM-RNN for this purpose. If the RNN is trained on the joint trajectory data from previous samples to predict the future joint trajectory, then it will be difficult to modify a large number of RNN inputs according to the need. One of the ways to work around this issue is to parameterize the joint trajectory in terms of its amplitude and frequency components. In that way, the number of inputs to the network will remain the same for different motion variations, which in turn will make the adaptation much simpler. We propose to use the fundamental DFT coefficients and the motion range to train the LSTM-RNN such that it can predict the suitable reference command for Zeno. A periodic discrete time signal S_k can be parameterized by its DFT coefficients as:

$$S_k = \sum_{l=0}^{N-1} S(l) \exp\left(\frac{-i2\pi kl}{N}\right) \quad (3)$$

where N is the number of samples, l is current sample, k is current frequency and $S(l)$ is the amplitude at sample l . Note that, predicting

a joint trajectory from only the fundamental DFT coefficient is a non-trivial task and the proposed LSTM-RNN based prediction will be more precise than approximating the future trajectories outright from the truncated DFT. For our deep learning network, five LSTM layers are used for processing the inputs, followed by five deep dense layers that generate the reference trajectories for the Zeno robot (see Fig. 5). The proposed network is trained with the fundamental DFT coefficients (a shape factor and a speed factor) computed from the subject's motion data and the motion range.

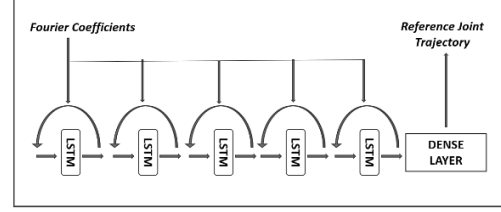


Figure 5: Proposed LSTM-RNN five-layer Architecture

In general, the RNN-based networks train on root mean square error (RMSE) between the predicted output and the actual output, or an entropy metric. As the objective of the proposed LSTM-RNN to adapt its output according to both the shape and speed of the subject's motion, we define a convex loss function (E_{Loss}) as:

$$\left(\sum_{j=1}^N \left(\|o_j - y_j\|^2 + \|(o_j - o_{j-1}) - (y_j - y_{j-1})\|^2 \right) \right) \quad (4)$$

where T is the time period of the joint trajectory, o_j is j_{th} output sample from the network and y_j represents the j_{th} sample of recorded subject data. The 1st term corresponds to shape error which is the same as a simple mean square error (RMSE). The second term captures the speed difference between RNN output and actual subject trajectory. The backpropagation through time for a layer ' l ' in the network can be expressed as the following set of equations:

$$\begin{aligned} \delta_h^{T,l} &= \nabla_{y^T} (E_{loss}) \sigma' \left(h_T^l \right), \quad \delta_c^{T,l} = \delta_h^{T,l} o_T^l \left(1 - \tanh^2 \left(c_T^l \right) \right) \\ \delta_o^{T,l} &= \delta_h^{T,l} \tanh \left(c_T^l \right) \sigma' \left(o_T^l \right), \quad \delta_f^{T,l} = \delta_c^{T,l} c_{T-1}^l \sigma' \left(f_T^l \right) \\ \delta_i^{T,l} &= \delta_c^{T,l} \tilde{c}_T^l \sigma' \left(i_T^l \right), \quad \delta_c^{T,l} = \delta_c^{T,l} i_T^l \left(1 - \tilde{c}_T^l \right)^2 \\ \delta_{W_i}^{T,l} &= \delta_i^{T,l} \left[h_{T-1}^{l-1}, h_T^{l-1}, x_T \right]^T, \quad \delta_{W_f}^{T,l} = \delta_f^{T,l} \left[h_{T-1}^{l-1}, h_T^{l-1}, x_T \right]^T \\ \delta_{W_o}^{T,l} &= \delta_o^{T,l} \left[h_{T-1}^{l-1}, h_T^{l-1}, x_T \right]^T, \quad \delta_{W_c}^{T,l} = \delta_c^{T,l} \left[h_{T-1}^{l-1}, h_T^{l-1}, x_T \right]^T \\ \delta_h^{t,l} &= \left(\delta_h^{t+1,l} W_i^l + \delta_i^{t+1,l} W_i^l + \delta_h^{t+1,l} W_f^l + \delta_f^{t+1,l} W_f^l \right) \sigma' \left(h_t^l \right) \\ \delta_c^{t,l} &= \delta_h^{t,l} o_t^l \left(1 - \tanh^2 \left(c_t^l \right) \right) \end{aligned} \quad (5)$$

where, the notation $\delta_h^{t,l}$ represents the gradient at time step t in layer l and $\delta_{W_i}^l$ terms represent weight updates for each gate in layer l .

Note that, when considering scalability in relation to the number of robot joints (N), each robot joint sequence introduces three additional inputs: shape factor, speed factor, and range. If we denote the number of robot joints as N , then the total number of inputs scales linearly with N , resulting in $3N$ inputs. Therefore, it is reasonable

to expect the training complexity to be approximately $O(N)$. This implies that training time is expected to increase linearly as we incorporate more robot joints into the system.

3 EXPERIMENTAL RESULTS

To validate the proposed AMI framework, we conducted experiments with 10 human subjects. The subjects were asked to follow a hand motion directed by our Zeno robot [7]. The same motion was repeated for three speed settings, i.e. Slow, Normal, and Fast. The Kinect camera recorded the motions of the subjects and then the camera data was processed to compute the motion waveform DFT and its SODTW cost when compared to the robot motion. Since the Zeno arm has four degrees of freedom, namely shoulder joints α and β , and elbow joints θ and γ (Fig. 7), we first calculated the percentage involvement of every joint in performing the arm motion. For the chosen motion, closely resembling a hammer exercise, the joint angle γ was dominant, and therefore our similarity SODTW cost was calculated based solely on this angle.

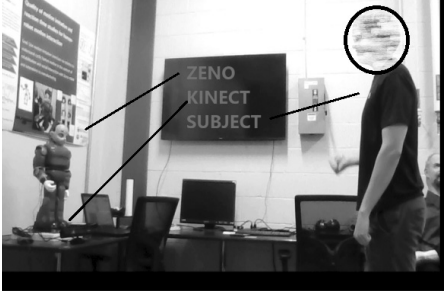


Figure 6: Imitation Experiment Setup

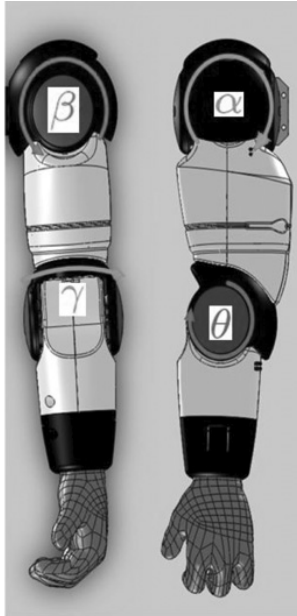


Figure 7: Zeno robot 4 degrees of freedom

Table 1: Weight (in terms of range) of joint angles contribution during a Hammer motion exercise

Joint Angle	Right-hand Hammer motion
α	0.073008126
β	0.055847682
γ	0.770277546
θ	0.100866647

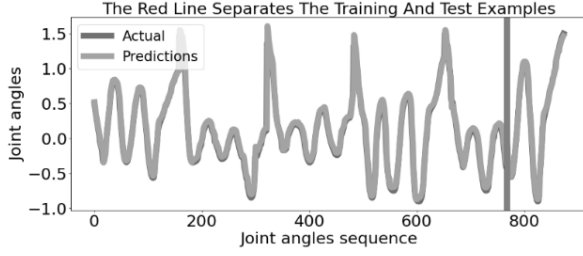
Table 2: SODTW cost stats for all subjects

Speed Mode	Average SODTW cost	Standard Dev.
Normal	36.8	2.9
Slow	65	5.3
Fast	10.9	0.9

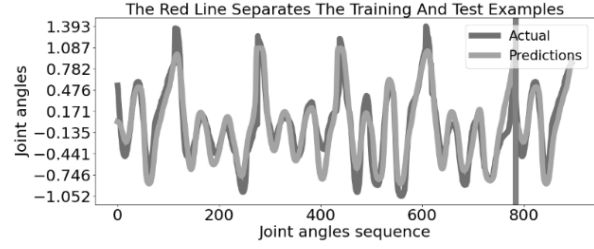
Further average SODTW cost is calculated for the 10 subjects imitating fast, slow, and normal types of motion. It can be observed that fast motion is easy to imitate whereas slow mode is most difficult for the subjects. Therefore, the magnitude, and frequency are reduced to make the imitation more challenging and vice versa.

The popular Python libraries TensorFlow and Keras were used for creating the deep networks. For the training process of AMI, we connected to an NVIDIA V100 TENSOR CORE GPU using an ASUS VivoBook Pro laptop equipped with an Intel Core i7 processor running at 2.80 GHz and 16 GB of RAM. During this setup, training typically ranged from 1 to 3 minutes, depending on the number of joint sequences and epochs. To verify that the proposed methodology works well to predict the future joint trajectory from Fourier coefficients, a baseline result was obtained. For that purpose, a simple LSTM-RNN with 1 LSTM layer and a dense layer (selected by trial and error to get a satisfactory result) was fed with the previous joint angles data and trained to predict the future joint trajectory. For this purpose, the data for each subject is normalized and scaled. Then the data is confined to a fixed number of samples (180). The simple LSTM-RNN is trained on 5 subjects and tested on a different subject data (after red line). The cost function for this baseline network is chosen as the RMS error between predicted and actual trajectories. The training error was found to be 0.018 and the test error was found to be 0.025 (see Fig. 8 (a)).

Once the baseline result was created, we used the same five subjects' data to compute their Fourier coefficients and the range of their motion. To train the network for various shapes and speeds of the subjects, we resample the subjects' data to increase/decrease Fourier coefficients (ref. Fig. 9). We used only the largest two fundamental components of amplitude as shape factor as well as frequency for our speed factor. We modulated the deep network architecture by changing the number of LSTM layers, dense layers, and learning rate till a comparable result as baseline RNN is obtained. The final architecture is shown in figure 5. For each subject, the entire trajectory data is divided into 3 cycles, each consisting of 60 samples. This fixed cycle length ensures consistent data processing. Note that, the inputs for each subject are kept the same Fourier coefficients and range of motion for all output samples. Training of

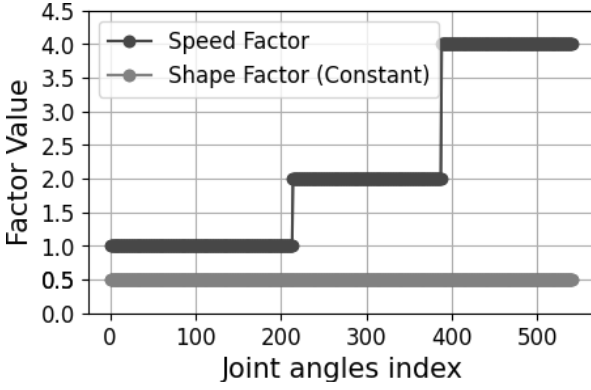


(a) Baseline LSTM-RNN

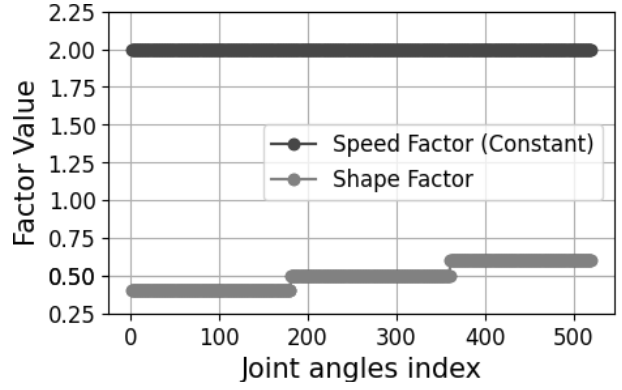


(b) LSTM-RNN Proposed Output

Figure 8: Prediction performance in our proposed networks



(a) Fourier Coefficients (shape and speed) for an Oversampled (slower), Normal and Undersampled (faster) Version of One Subject Data with Fixed Range



(b) Fourier Coefficients (shape and speed) for a lower range, normal, and higher range motion of One Subject Data with Fixed Speed

Figure 9: DFT Computation

our network is performed using the popular Adam optimizer, which updates the network weights through backpropagation in time (see Equation 5)). The Adam optimizer is well-suited for optimizing convex and stochastic loss functions, utilizing first-order gradient descent. The loss during training is found to be 0.05 and during testing is found to be 0.08 (see Fig. 8 (b)).

To test our framework for AMI, we computed the SODTW cost for each subject. We chose different subjects and instructed them to deliberately not follow Zeno's motion perfectly. We chose different subjects where we asked the subjects to move slower, faster, lower range, and higher range. The reference trajectory generated from the LSTM- RNN is presented in comparison to the subject's joint trajectory (ref Figs. 10 and 11). The RMSE and speed errors are presented in Table 3 and figure 13. For one specific test subject, we asked them to perfectly follow the hammer motion, resulting in a small SODTW cost of 21 (lower than the average). We used this cost and range as inputs to our Fuzzy engine, which recommended increasing the amplitude of Fourier coefficients by a factor of 1.3 and decreasing the frequency by a factor of 0.7 (increase difficulty level). The inputs to the network were adjusted accordingly to implement these modifications, and the network's output is shown in figure 12. The RMSE for shape and speed error was found to be 0.005 and 0.003 respectively.

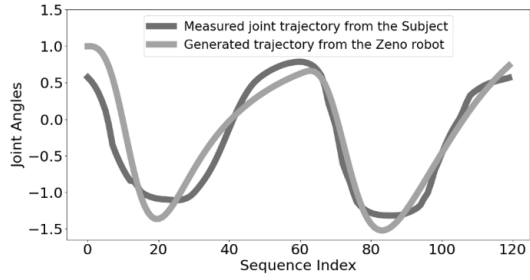
Table 3: Prediction Accuracy for Motion Variation

Task Variation	Shape RMSE	Speed RMSE
Slower Speed	0.006	0.005
Higher Speed	0.005	0.004
Lower Range	0.003	0.002
Higher Range	0.003	0.004
Higher Range + Slower Speed	0.005	0.003

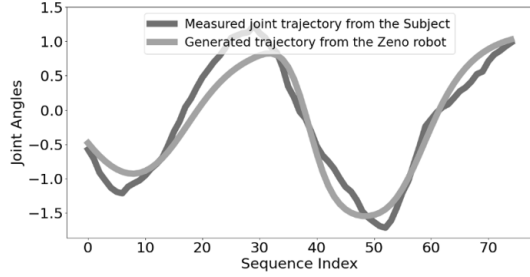
In summary, our proposed Adaptive Motion Imitation (AMI) framework demonstrates the ability to adapt accurately to both speed and shape changes in subjects' motions, as evidenced by the experimental results. Moreover, our framework excels in scenarios where subjects' motions exhibit simultaneous changes in both speed and shape, further highlighting its versatility and effectiveness.

4 CONCLUSION AND FUTURE WORK

In this paper, we proposed a modular AMI framework for robot-assisted physiotherapy for subjects with Autism Spectrum Disorders (ASD). The framework uses a deep LSTM-RNN to predict the reference joint trajectory for our Zeno robot when fed with the

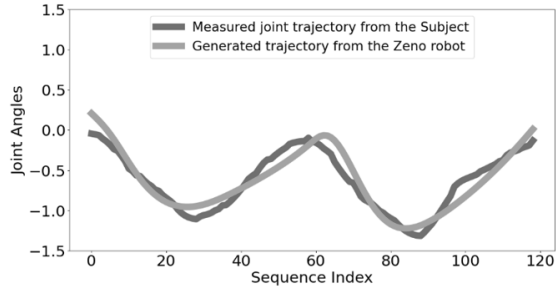


(a) Subject performing the motion with a slow speed (0.7x)

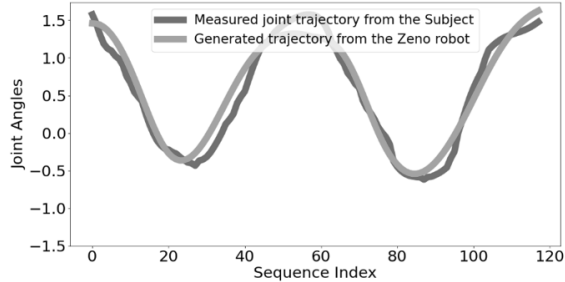


(b) Subject performing the motion with a fast speed (1.3x)

Figure 10: Generated sequence after training for the subject performing the motion with different speeds (the reduced number of sequence index in the faster motion stems from the under sampling of the data)



(a) Subject performing the motion with a lower range (0.9x)



(b) Subject performing the motion with a higher range (1.2x)

Figure 11: Generated sequence after training for the subject performing the motion with different ranges

Fourier coefficients and range of motion for a given human subject. The network exploits the SODTW algorithm to modulate the inputs

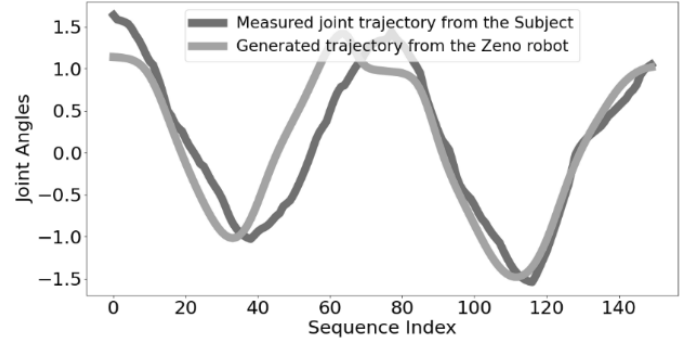


Figure 12: Generated sequence after training for the subject performing the motion with a high range and slow speed.

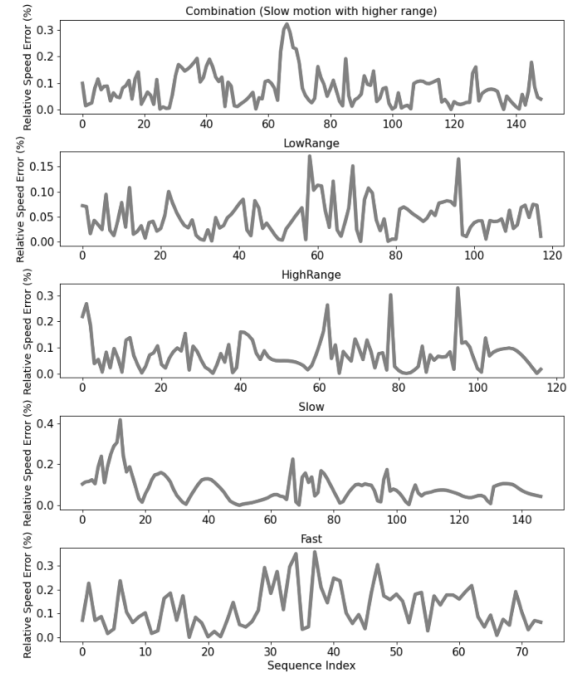


Figure 13: Speed Error for Different Motion Variations

according to the ability of the subject to imitate Zeno's behavior, thus building a fuzzy inference engine. A study conducted on 10 adult subjects proved the validity of our AMI framework by predicting appropriate robot exercises resembling human subject motions. Our overall AMI framework scales favorably and can be extended to motions involving more degrees of freedom and longer motion sequences.

In future work, we intend to carry out more extensive experimentation with this framework, including testing it on children with ASD and different physiotherapy programs.

REFERENCES

- [1] R. M. Andrade, P. H. F. Ulhoa, and C. B. S. Vimieiro, "Designing a highly backdrivable and kinematic compatible magneto-rheological knee exoskeleton," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5724–5730.

- [2] E. Nazzi, E. Canzi, G. Piga, A. Galassi, G. Lippi, and G. Benassi, "Segment online dtw for smart rehabilitation of asd children: A preliminary study," in *Proceedings of the 4th EAI International Conference on Smart Objects and Technologies for Social Good (GOODTECHS)*, 2015, pp. 122–126.
- [3] A. Frolov, I. Kozlovskaya, E. Biryukova, and P. Bobrov, "Use of robotic devices in post-stroke rehabilitation," *Neuroscience and behavioral physiology*, vol. 48, pp. 1053–1066, 2018.
- [4] T. Goyal, S. Hussain, E. Martinez-Marroquin, N. A. T. Brown, and P. K. Jamwal, "Stiffness-observer-based adaptive control of an intrinsically compliant parallel wrist rehabilitation robot," *IEEE Transactions on Human-Machine Systems*, vol. 53, no. 1, pp. 65–74, 2023.
- [5] M. Kirtay, J. Chevale're, R. Lazarides, and V. V. Hafner, "Learning in social interaction: Perspectives from psychology and robotics," in *2021 IEEE International Conference on Development and Learning (ICDL)*, 2021, pp. 1–8.
- [6] H. Mahdi, S. A. Akgun, S. Saleh, and K. Dautenhahn, "A survey on the design and evolution of social robots — past, present and future," *Robotics and Autonomous Systems*, vol. 156, p. 104193, 2022.
- [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," in *Int. J. Soc. Robot.*, vol. 8, no. 5, 2016, p. 695–707.
- [8] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004, pp. 1–8.
- [9] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," in *Robotics and Autonomous Systems*, vol. 57, no. 5, 2009, pp. 469–483.
- [10] A. Hussein, M. M. Gaber, and E. Elyan, "Imitation learning: A survey of learning methods," in *Artificial Intelligence Review*, vol. 48, no. 1, 2017, pp. 31–60.
- [11] Z. Zheng, E. M. Young, A. R. Swanson, A. S. Weitlauf, Z. E. Warren, and N. Sarkar, "Robot-mediated imitation skill training for children with autism," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 6, pp. 682–691, 2016.
- [12] W. Liu, L. Peng, J. Cao, X. Fu, Y. Liu, Z. Pan, and J. Yang, "Ensemble bootstrapped deep deterministic policy gradient for vision-based robotic grasping," *IEEE Access*, vol. 9, pp. 19 916–19 925, 2021.
- [13] P. Sharma, D. Pathak, and A. Gupta, "Third-person visual imitation learning via decoupled hierarchical controller," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.
- [14] S. Park, J. H. Park, and S. Lee, "Direct demonstration-based imitation learning and control for writing task of robot manipulator," in *2023 IEEE International Conference on Consumer Electronics (ICCE)*, 2023, pp. 1–3.
- [15] W. Xu, J. Huang, Y. Wang, C. Tao, and L. Cheng, "Reinforcement learning based shared control for walking aid robot and its experimental verification," *Advanced Robotics*, vol. 29, no. 22, pp. 1463–1481, 2015.
- [16] J. Xu, L. Xu, Y. Li, G. Cheng, J. Shi, J. Liu, and S. Chen, "A multi-channel reinforcement learning framework for robotic mirror therapy," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5385–5392, 2020.
- [17] S. S. P. A. Bishe, T. Nguyen, Y. Fang, and Z. F. Lerner, "Adaptive ankle exoskeleton control: Validation across diverse walking conditions," *IEEE Transactions on Medical Robotics and Bionics*, vol. 3, no. 3, pp. 801–812, 2021.
- [18] N. Taghavi, M. H. A. Alqatamin, and D. O. Popa, "Ami: Adaptive motion imitation algorithm based on deep reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 798–804.
- [19] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv Preprint arXiv:1506.00019*, 2015.
- [20] S. Yang, W. Zhang, R. Song, J. Cheng, H. Wang, and Y. Li, "Watch and act: Learning robotic manipulation from visual demonstration," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.
- [21] M. Doering, D. F. Glas, and H. Ishiguro, "Modeling interaction structure for robot imitation learning of human social behavior," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 219–231, 2019.
- [22] H. Kim, Y. Ohmura, A. Nagakubo, and Y. Kuniyoshi, "Training robots without robots: deep imitation learning for master-to-robot policy transfer," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2906–2913, 2023.
- [23] S. Yang, W. Zhang, R. Song, W. Lu, H. Wang, and Y. Li, "Explicit-to-implicit robot imitation learning by exploring visual content change," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 4920–4931, 2022.
- [24] D. X. V. R. Johnson-Roberson, "M bio-lstm: a biomechanically inspired recurrent neural network for 3-d pedestrian pose and gait prediction ieec robot," *Autom. Lett.*, vol. 4, no. 2, p. 1501, 2019.
- [25] K. Kawaharazuka, Y. Kawamura, K. Okada, and M. Inaba, "Imitation learning with additional constraints on motion style using parametric bias," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5897–5904, 2021.
- [26] N. Taghavi, J. Berdichevsky, N. Balakrishnan, K. C. Welch, S. K. Das, and D. O. Popa, "Online dynamic time warping algorithm for human-robot imitation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3843–3849.