
IDL Midterm Report

Changseob Song*, Maria Tagliaferri*, Max Song*, Riven Lee*
Carnegie Mellon University
Pittsburgh, PA 15213

1 Introduction

Robotic lower limb exoskeletons have demonstrated significant potential to improve community ambulation and general ease of locomotion for a wide range of user populations. These wearable devices provide assistance to one or more joints of the user's leg by applying a specified torque to the joint via small actuators. This robotic assistance reduces the energetic cost of walking and increase user stability while navigating various environments. Assistance concentrated at the hip joint has been found to significantly reduce energy expenditure and gait abnormality compared to other joints [1]. To be effective, however, the magnitude and frequency of the provided assistive torque must be optimized based on the user's kinetic state. Joint moment and gait phase are the two most important states in the context of exoskeleton assistance. Estimating joint moments of the lower body allows for real-time improvements to exoskeleton controller decisions regarding this profile of assistance. This complex estimation task can be made efficient with the use of deep learning models such as convolution neural networks (CNNs). Temporal Convolution Networks (TCNs), a variation of CNNs that are designed specially for time series data, are often chosen for this application due to their stability and efficiency. In this paper we propose a novel adaptation of the Temporal Convolution Network for hip joint moment estimation developed by Dean D. Molinaro et al. at the Georgia Institute of Technology [2], [3].

2 Literature Review

Machine learning models have been widely used for estimating key user state information, such as gait cycle [4], joint angles [5], and joint moments [1]. While gait phase and joint kinematics serve as important indicators of the locomotion state, joint kinetics play the most significant role due to their direct coupling with exoskeleton torque control. Accurately estimating the instantaneous joint moment ensures that the exoskeleton provides sufficient torque to the hip joint at each phase of the gait cycle. Latency or error in this estimation can result in suboptimal profiles of torque generation and negative impacts the user. Providing torque at the wrong time in the gait cycle or at too high a magnitude can cause physical injury to the user. Conversely, providing too little torque increases energetic cost of locomotion and the risk of gait instability. Given the importance of accurate biological state estimation, studies have deployed various machine learning methods for this task including long short-term memory (LSTM) [6], convolutional neural network (CNN) [7], and TCN [1].

LSTMs were used in early explorations of joint moment estimation because they exhibit robust performance on time-series data and their long-term memory allows for the capture of periodic gait patterns. However, further developments in the field revealed that their architectural complexity and subsequent high computational cost limit the model's performance on real-time data [6]. Popularity began to shift towards the use of CNNs due to their high computational efficiency and generalizable architecture [7]. These characteristics allowed for effective real-time estimation and customization of

*Division of Labor: Changseob Song: baseline code development and data processing, visualization Maria Tagliaferri: literature review and baseline code development Max Song: baseline code development and model implementation, Riven Lee: data processing and model implementation.

the model for various locomotion tasks. Despite these benefits, CNNs struggled to capture long-range dependencies of sequential data. This functionality is essential for biological state estimation given the highly cyclical nature of gait kinetics and the subsequent importance of understanding a user’s long-term gait patterns. In 2021, researchers at Jilin University proposed a combined LSTM-CNN approach which attempted to leverage the benefits of both architectures with IMU sensor inputs. The LSTM was used to extract temporal features of gait data, while the CNN extracted spatial features [8]. This novel approach demonstrated high performance and low latency in gait phase estimation, however its performance did not exceed that of CNN or LSTM models in regards to joint moment estimation.

Adapting this combined model approach, researchers at Georgia Institute of Technology utilized a TCN model to perform the task of joint moment estimation [2]. The structure of the TCN leveraged temporal data from a single IMU sensor as a substitute for multijoint sensing, resulting in accurate hip moment estimates that are generalizable across subjects. A TCN was chosen due to its strengths in capturing both long and short term dependencies as well as its parallelism which allows for low latency. In this way, TCNs have similar beneficial characteristics of both LSTMs and CNNs. Here, we validate and adapt this TCN architecture to estimate biological joint torque the goal of increasing accuracy and decreasing latency.

3 Dataset Description

The novel adaptation of the TCN we propose will be developed and benchmarked on the dataset released by Dean D. Molinaro et al. [3] This dataset includes biomechanical data collected from 34 participants in 4 distinct phases of the experiment. Participants wore the exoskeleton and data was collected using onboard sensors as well as motion capture (Ground Truth). Participants were asked to adopt a broad range of ambulation modes including 1) LG: level ground (either overground or on a treadmill) 2) RA: ramp ascent (either overground or on a treadmill) 3) RD: ramp descent (either overground or on a treadmill) 4) SA: stair ascent (overground staircase) 5) SD: stair descent (overground staircase) 6) ST: standing 7) TR: stand-to-walk and walk-to-stand transitions in same trial (treadmill) 8) TRA: stand-to-walk transition (overground) and 9) TRB: walk-to-stand transition (overground). Inputs to our model include the ipsilateral (same side of body) actuator encoder position and velocity, six-axis ipsilateral thigh-mounted IMU (Inertial Measurement Unit) data, and six-axis pelvis-mounted IMU data windowed over an appropriate time range. The output of the model will be compared with joint moment ground truth to gauge the accuracy and effectiveness of our adaptations.

4 Baseline Model

Our baseline Temporal Convolutional Network (TCN) model for hip moment estimation builds upon the work of Dean D. Molinaro et al. [2]. The model consists of five residual blocks, each comprising two one-dimensional causal convolutional layers. Each layer is followed by a weight normalization layer and a ReLU activation function. The dilation factor increases exponentially every second layer. Each residual block is comprised of two 1D convolutional layers, each followed by a weight normalization layer and a ReLU activation function. This structure is shown in the expanded section in Figure 1. A simplified graphic depicting the effect of convolutional kernel dilation is also shown in Figure 1, in which x_t denotes the input data at time t , and $\hat{y}_t$ denotes the corresponding model output at the same time step. A 930ms window of sensor data was used to estimate hip moment in both the flexion and extension axes at each time instance.

As shown Figure 2, hyperparameter optimization was executed on the baseline model. The number of layers, hidden size, dropout probability, input window size, and learning rate of the TCN were optimized. Additionally, batch normalization after the hidden layers of the model was also tested for each combination of hyperparameters. Based on pilot results, the initial window size and learning rate were fixed to 500 ms and 0.0001, respectively, to maintain a tractable number of hyperparameter combinations

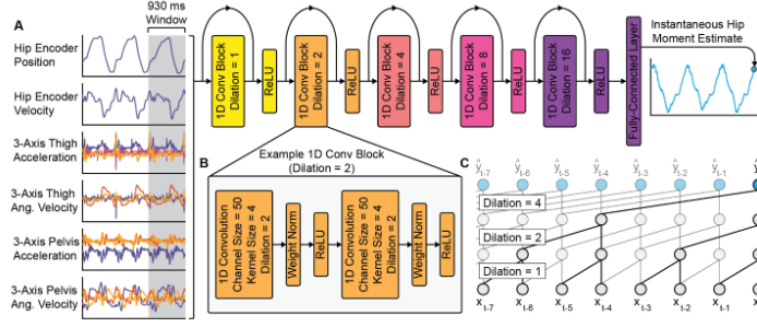


Figure 1: Baseline Model Diagram

TABLE IV
TCN HYPERPARAMETER OPTIMIZATION

Hyperparameter	Candidate Values	Selected Value
# of Levels	1, 2, 3, 4, 5	5
# of Channels per Hidden Layer	10, 30, 50, 80	50
Kernel Size	4, 5, 7, 10, 20	4
Dropout Probability ^a	0, 0.1, 0.2, 0.3, 0.4, 0.5	0.3
Learning Rate ^a	1e-4, 5e-4, 1e-3	5e-4
# of Learnable Parameters	N/A	94,601

Figure 2: Baseline Hyperparameter Optimization

5 Baseline Model Implementation

The baseline model was implemented using the provided open source dataset as described in section 3. The model architecture is described in Figure 1, and was optimized by a hyperparameter sweep as tabulated in Figure 2. When originally trained with the optimized hyperparameter values, our implementation achieved a root mean square error (RMSE) of 0.4 Nm/kg. This was about 3 times the value of 0.131 Nm/kg as achieved by the TCN in the original paper [Figure 4]. We then executed an additional sweep of hyperparameters as a first attempt to better match the results of the baseline model. However, we were unsuccessful in significantly decreasing the RMSE below our original value. As shown in Figure 3, we next performed k-fold cross validation to try to improve our validation RMSE. Given that gait variability is common between subjects, we predicted that a k-fold validation using

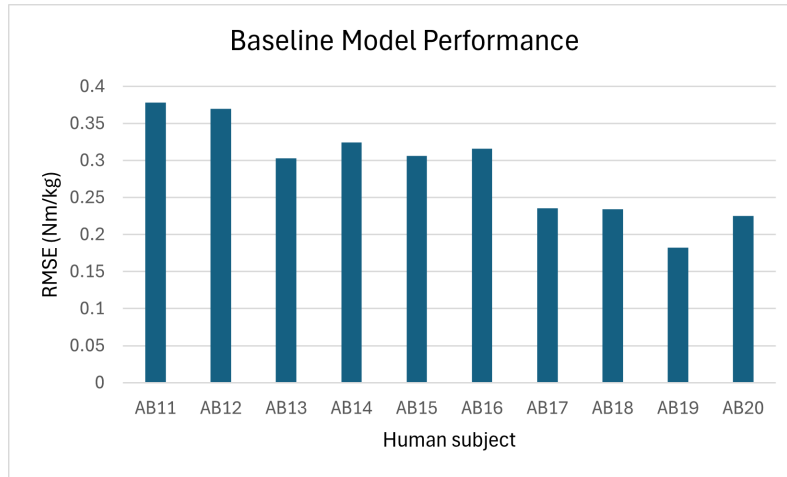


Figure 3: Baseline Model Training Performance

HIP MOMENT ESTIMATION RESULTS FOR STEADY-STATE AMBULATION				
	Our Approach (TCN)	FCNN	LSTM	Baseline Method
RMSE (Nm/kg)	0.131±0.018*	0.150±0.024*†	0.149±0.020*†	0.173±0.015†
R ²	0.880±0.030*	0.848±0.039†	0.842±0.037†	0.793±0.061†

* and † represent a statistical difference from the baseline method and TCN estimates, respectively (p<0.05). All other comparisons were not statistically different. Best performance according to RMSE and R² is bolded. Results are presented as the leave-one-subject-out average ± 1 standard deviation.

Figure 4: Baseline Model RMSE

a single subject dataset at a time as the testing fold would yield improved results. We were able to reduce our RMSE significantly, to about 16.5 Nm/Kg using this method. Our implementation therefore performed very closely to the original model, with a slight difference in performance. One potential cause of our slightly reduced performance is data interpolation that was executed due to the prevalence of Nan values in the label dataset. As described by the authors, the Nan values indicate time windows when subjects stepped beyond the force-plate and therefore no joint moment data could be collected [2]. The authors do not explicitly state the data processing method they used to navigate these frequent Nan values, therefore we made an informed choice to perform interpolation to replace these values. This interpolation may be impacting the performance of our implementation, therefore we will reevaluate optimal data processing or choose a different dataset in our future adaptations of the baseline.

6 Proposed Model Extensions

The baseline model can be improved upon in the following ways 1) since the baseline model was originally intended to run on a low-powered embedded device, its performance is potentially limited by a low parameter count and relatively shallow network architecture. We propose to extend the baseline model by using deeper architecture, large kernel and filter sizes, which can better take advantage of the next generation of more capable embedded AI processors. 2) TCN was selected as the baseline model due to research which indicates it's surprising effectiveness at sequence modeling tasks. However, this does not mean we should overlook dedicated sequence modeling architectures, such as TDNN, RNN, LSTM, and transforms. We propose to improve on the baseline model's architecture selection by exploring the suitability of these sequence models. 3) Besides, established sequence models, novel models developed in the latest research may be suitable for our task. Kolmogorov–Arnold Networks (KAN) is an extension of MLP which has demonstrated improved performance on AI + Science tasks with an added benefit of being interpretable. We intend to experiment with this novel architecture as an extension of the baseline model.

References

- [1] Pan YT. et al., Effects of Bilateral Assistance for Hemiparetic Gait Post-Stroke Using a Powered Hip Exoskeleton. Ann Biomed Eng. 2023 Feb; 51(2):410-421.
- [2] Dean D. Molinaro et al. ,Estimating human joint moments unifies exoskeleton control, reducing user effort.Sci. Robot.9,eadi8852(2024).
- [3] Molinaro, Dean (2024). Estimating human joint moments unifies exoskeleton control and reduces user effort [Dataset]. Dryad.
- [4] Kang et al., Real-Time Gait Phase Estimation for Robotic Hip Exoskeleton Control During Multimodal Locomotion, IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 3491-3497, April 2021.
- [5] Lee, T., Kim, I., Lee, S.-H., Estimation of the Continuous Walking Angle of Knee and Ankle (Talocrural Joint, Subtalar Joint) of a Lower-Limb Exoskeleton Robot Using a Neural Network. Sensors 2021, 21, 2807.
- [6] J. Li et al., "A Novel Method of Pattern Recognition Based on TLSTM in lower limb exoskeleton in Many Terrains," 2022 4th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), Hangzhou, China, 2022, pp. 733-737.

[7] U. Côté Allard et al., "A convolutional neural network for robotic arm guidance using sEMG based frequency-features," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea (South), 2016, pp. 2464-2470.

[8] Chen, Cf., Du, Zj., He, L. et al. A Novel Gait Pattern Recognition Method Based on LSTM-CNN for Lower Limb Exoskeleton. J Bionic Eng 18, 1059–1072 (2021). <https://doi.org/10.1007/s42235-021-00083-y>