Deep Learning-based Task-Agnostic Controller for Ankle Exoskeleton

Divyansh Gupta
Northeastern University
gupta.divy@northeastern.edu

Abstract

Lower-limb wearable devices like exoskeletons and prosthetics are limited to lab settings because of their inability to work in natural environments, which require modulating assistive torque profiles to adjust for different joint dynamics across various ambulation modes or tasks. In this study, we provide a subject and task-independent control strategy for the ankle exoskeleton that uses deep learning to predict biological ankle torque which linearly maps to exoskeleton assistive torque. IMU, goniometer, and pressure insole data from 3 subjects across five different tasks are used to train traditional deep-learning models, including fully-connected neural networks (FCNN), long short-term memory (LSTM) networks, and temporal convolution neural (TCN) networks. Evaluation of these neural networks on an unseen subject in training across 22 hold-out tasks concluded that TCN outperforms other neural nets for this application.

1. Background

Lower-limb robotic wearable devices have shown great potential to improve people's quality of life through enhanced mobility - reducing the metabolic cost of walking and improving gait functions. To enable their use among healthy populations, it becomes necessary to have a control policy that supports multimodal ambulation (e.g., different walking speeds, running, inclined walks, jumping, etc.) and allows smooth transition across each mode. In most available wearable devices, a three-level control system is often employed, consisting of high-level, mid-level, and low-level controllers. The high-level controller functions as a classifier to determine the ambulation mode. Subsequently, the mid-level controller generates a reference assistance trajectory based on the classified mode, and the low-level controller is responsible for tracking this trajectory. This control system hierarchy enables efficient and accurate control of wearable devices in continuous ambulation mode with finite switching modes. However, this high-level control policy may not be suitable in natural settings where ambulation modes are not steady-state modes and cannot be discretized.

In order to address the aforementioned issue, some researchers have employed biological torque estimators as high-level controllers in exoskeletons [5]. Various techniques have been utilized to develop multiple torque estimators, including neuromuscular-based forward dynamics, joint kinematics and kinetics-based inverse dynamics, and data-driven estimators. The inverse dynamics method is considered the most widely accepted and straightforward of the three methods [2]. This approach utilizes data from motion capture and force plates to estimate joint torque employing Equation 1. However, the inverse dynamics-based method has certain limitations, rendering it unsuitable for real-time applications outside the lab environment and yielding inaccurate results with wearable sensors.

One of the primary limitations of the inverse dynamics-based method is its reliance on accurate ground reaction forces (F and Fv) and center of pressure (x) location about the ankle-joint axis. Unfortunately, wearable sensors such as pressure insoles only provide the vertical component (Fv), making it impossible to obtain an accurate estimation [1]. Furthermore, the method requires a precise measure of angular acceleration, often noisy when calculated using a double-derivative of ankle angle (θ) estimated using Goniometers. In addition, it cannot be generalized to subject-independent approaches, as the inertia (I) and ankle-joint axis height (d) from the foot base can vary for each user. Consequently, these limitations have restricted using inverse dynamics in real-time outdoor systems such as exoskeletons.

$$\tau = F_V \cdot x \cdot \sin\theta + I \cdot \frac{d^2\theta}{dt^2} + F \cdot d \tag{1}$$

Recent computational resource advancements have made data-driven techniques increasingly popular for torque estimation. These techniques require less sensor information and can lead to subject-independent generalized models with accurate results. For example, [6] utilized XGBoost and FCNN to estimate biological hip torque using mechanical sensor data from a hip-exoskeleton and [7] used CNN to estimate full-lower body biomechanics for walking and

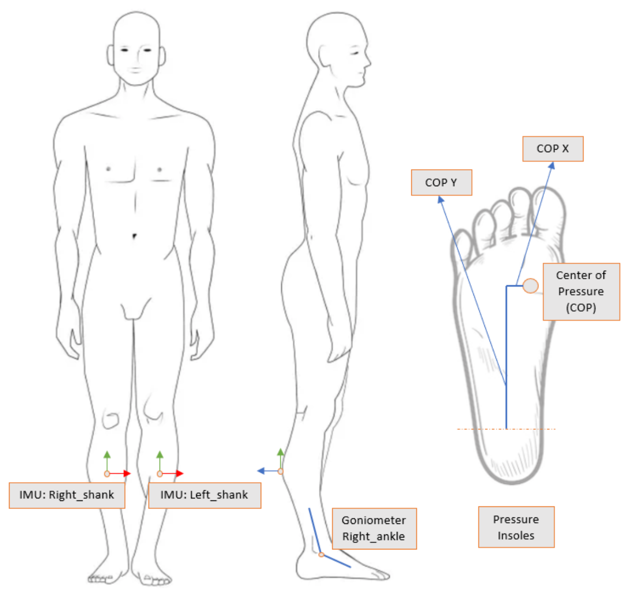


Figure 1. Sensors used for selected features - IMUs (Gyroscope, Accelerometer), Electric Goniometer and Pressure Insoles

running. Similarly, [7] used FCNN to estimate knee torque using inertial sensors. Moreover, Molinaro *et al.* applied this data-driven biological torque estimation technique to drive the hip and knee exoskeleton in their study [5]. It is important to note that ankle joint torque profiles vary widely across ambulation modes and subjects. Therefore, this study aims to investigate the potential of data-driven techniques for ankle exoskeletons. We also seek to determine the minimum number of ambulation modes and sensors required to establish a generalized model.

2. Dataset

For our study, we utilized a comprehensive dataset consisting of wearable sensor data for the entire lower body of five participants across 27 different ambulation modes/tasks, with various variations of a subset of modes. Each mode in the dataset contains a minimum of 4000-time steps, corresponding to an average of 20 seconds of data. The dataset includes 80 features for each sample point, representing data from sensors such as IMU, goniometer, EMG sensor, and pressure insoles. This rich and diverse dataset allowed us to thoroughly evaluate the potential of data-driven techniques for accurate torque estimation in ankle exoskeletons.

To develop a generalized model capable of accurately predicting biological torque for unseen subjects and tasks, we partitioned the dataset into training, validation, and testing sets. The training dataset consists of data from three subjects performing five tasks, while the validation dataset comprises data from the same three subjects, but on 22 hold-out tasks that were not seen during training. Finally, the testing dataset contains data from one subject who was not involved in the training phase, and it includes data from 22 hold-out tasks. This approach enabled us to evaluate the performance of the model on unseen tasks and subjects while ensuring that the model is generalizable.

2.1. Forward Selection - Ambulation Modes

For determining the optimal subset of tasks for model training, we employed the forward selection technique with the FCNN model and further refined the selection based on the repeatability of the selected tasks within the experimental setup. Figure 2 presents the changes in MSE values on the validation dataset as each ambulation mode is added to the training model during the forward selection process. Our analysis revealed that a set of five tasks provided representative and diverse data for accurate torque estimation. Specifically, the tasks included front-back jump, 90-degree turning jump, running at 2.5 m/sec, shuffle-walk, and dynamic walking with high knees. While it is important to note that this specific set of tasks was chosen based on their suitability for experimental setup, we recognize that these tasks are relatively difficult to repeat and not commonly used in other studies. In other studies, tasks such as walking, running, jumping, ramp-walk, and stair climbing are commonly employed for model training in this field. Additionally, we acknowledge the importance of considering the potential variability in task performance across different individuals and environments, which may impact the generalizability of the developed model.

2.2. Feature Selection

We conducted experiments with different feature sets, ranging from using all sensors - 80 features to limited sensors (2 IMUs and 2 goniometers) - 14 features. We hypothesized that a subset of features contributing directly to the torque estimation equation (equation 1) should be more relevant than others. This subset of features includes pressure insoles data, ankle angle, shank accelerometer data, and EMG data of soleus and gastrocnemius muscles. To validate this hypothesis, we trained a regression model and examined the f-values of each feature. Moreover, we iteratively selected and deselected the hypothesized relevant features to evaluate the FCNN model's performance on the validation data. Despite the high f-value and relevance of the EMG sensor, it was discarded as the data was found to be very noisy and may lead to incorrect results when used with real hardware. To refine our feature selection, we chose features corresponding to sensors commonly used with ankle exoskeletons. Eventually, 19 features were selected for training the model corresponding to data from IMUs at the shank, ankle angle, and pressure insoles. Reducing the number of features used for training did not have a signif-

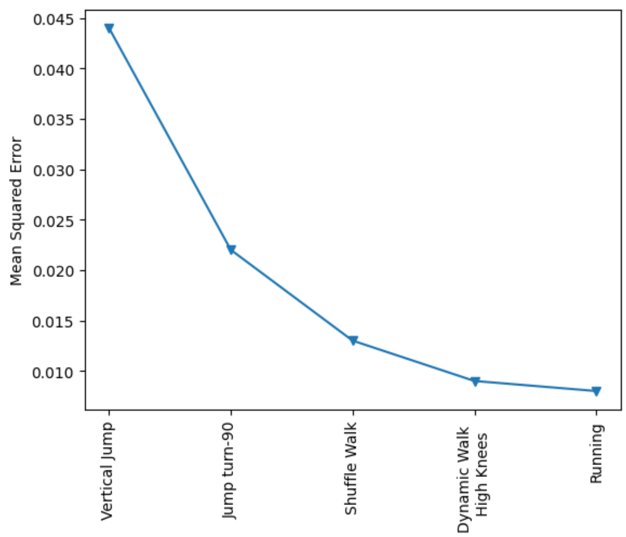


Figure 2. Changes in MSE on hold-out tasks with each added task for training the model

icant impact on the model's performance, which supports our hypothesis. Our findings highlight that selecting features carefully is critical for developing efficient machine learning models, and domain knowledge should be considered when selecting features for any machine learning model.

2.3. Data Preprocessing

The structure of our dataset involved individual feature files for each task and subject so to generate a single training dataset, it was important to ensure data continuity during model training especially when processing temporal information with RNN and TCN models. When loading and stacking data for each subject and task, we made sure to truncate sample points using the batch size and time-sequence length. This helped us avoid discontinuities within each training batch. After truncation, we performed min-max scalar transformation on the entire training dataset since the features represent physical quantities constrained by the human body. Furthermore, to prevent any information leak from training to testing or validation data, we applied the min-max scalar transformation parameters of the training model to testing and validation data rather than estimating them separately. This approach ensures that the data is standardized and consistent across all subjects and ambulation modes, reducing the risk of bias in our results.

3. Methodology

Our study employs three distinct Machine Learning models to estimate ankle joint torque, with each model implemented using the deep learning framework, TensorFlow 2.10.0. The hyperparameters and overall architecture for each model were chosen iteratively through experimentation. Furthermore, each model was compiled using Adam optimizer with MSE loss function. Additionally, a callback function was added while fitting each model to ensure the best-performing model on the validation dataset is saved. The following subsections present a detailed architecture and hyperparameters for each model.

3.1. FCNN

The FCNN, or fully-connected neural network, is a deep learning model that uses a series of hidden nodes activated by non-linear functions. This introduced non-linearity enables the model to capture non-linear relationships between input features and output labels. In recent years, the FCNN has been used in various research studies, including [5] and [7], to estimate biological joint torques. In our study, we also employed the FCNN model to estimate ankle joint torque from wearable sensor data. Unlike in a similar study by Molinaro *et al.* [5], we did not include any feature engineering layer to capture temporal information as we believed the underlying physical model for joint dynamics could be independently used at each time step. We experimented with various hyperparameters and found that an optimal architecture for our model was achieved with four hidden layers, with 128, 64, 32, and 16 neurons, respectively, and ReLU as the activation function. This activation function was used in each layer, allowing for efficient computation and accurate prediction of ankle joint torque.

3.2. LSTM

The long-short-term memory (LSTM) model is a recurrent neural network well-suited for modeling sequential data and has shown promising results in estimating biological torques [5]. red to traditional RNNs, LSTM has the advantage of remembering longer sequences without dealing with the vanishing gradient problem due to its gated architecture. Input training data for our model was reshaped in [Batch size, Time-sequence window, Features] format. In our model, we used an overlapping window, where each time-sequence window partially overlaps with the previous one, to capture the temporal dependencies in the data. The overlapping window size and the time-sequence length were chosen as hyperparameters and were set to 3 and 50, respectively. The LSTM model consisted of 2 LSTM layers with sizes of 50 nodes each, followed by a Dense output layer. Considering the size of the training and validation dataset, our model needed to run on CUDA so we decided to stick with the compatible activation function ”Tanh” for LSTM layer while the output layer used a linear activation function.

Model	HO-Tasks	HO-Tasks Exo
FCNN	0.126	0.098
LSTM	0.126	0.097
TCN	0.111	0.093

Table 1. RMSE (N.m/Kg)- TCN gives best results

3.3. TCN

The Temporal Convolutional Network (TCN) [3] is a 1D convolutional neural network well-suited for processing sequential data. Using dilated convolutions, TCNs can effectively capture temporal dependencies in the data and model long-range dependencies more efficiently than traditional recurrent neural networks. This makes them a computationally efficient and easy-to-train alternative to RNNs. However, because TCNs rely on convolutional layers, they may struggle with capturing complex patterns in the data. One way to improve the performance of TCN is to use Residual blocks, essentially shortcut connections between layers, which can help the network learn more complex patterns effectively. Our proposed TCN architecture utilizes 4 residual blocks with increasing dilation rates of (1,4,16,64) and a kernel size of 4. Additionally, each block has 50 filters with causal padding, which allows for effective modeling of the data's temporal dependencies while minimizing the risk of overfitting.

4. Results and Analysis

The Machine Learning models were evaluated using the root mean squared error (RMSE) metrics, which were computed for each ambulation mode separately and then averaged across all modes to obtain a global performance measure for each model. The average RMSE values for all holdout-ambulation modes and all holdout-ambulation modes relevant to an ankle exoskeleton are shown in Table 1. The TCN model achieved the lowest error compared to the LSTM and FCNN models, with an average RMSE of 0.093 N.m/Kg. The LSTM model had a slightly higher RMSE with an average of 0.097, while the FCNN model had the highest RMSE with an average of 0.098. These results demonstrate that the TCN model outperformed both the LSTM and FCNN models in terms of accuracy in predicting ankle joint torque.

The lower RMSE achieved by the TCN model can be attributed to its ability to capture long-term dependencies in the data and effectively model non-linear relationships between the input features and output labels. While the LSTM model is also designed to capture long-term dependencies, it may have been less effective than the TCN due to its difficulty in capturing long-term dependencies. Most of the tested tasks were cyclic and had repeatable patterns, which might have been captured by TCN effectively compared to LSTM. On the other hand, the FCNN model may not have captured the sequential nature of the data as effectively, as it lacks the recurrent connections of the LSTM and TCN models.

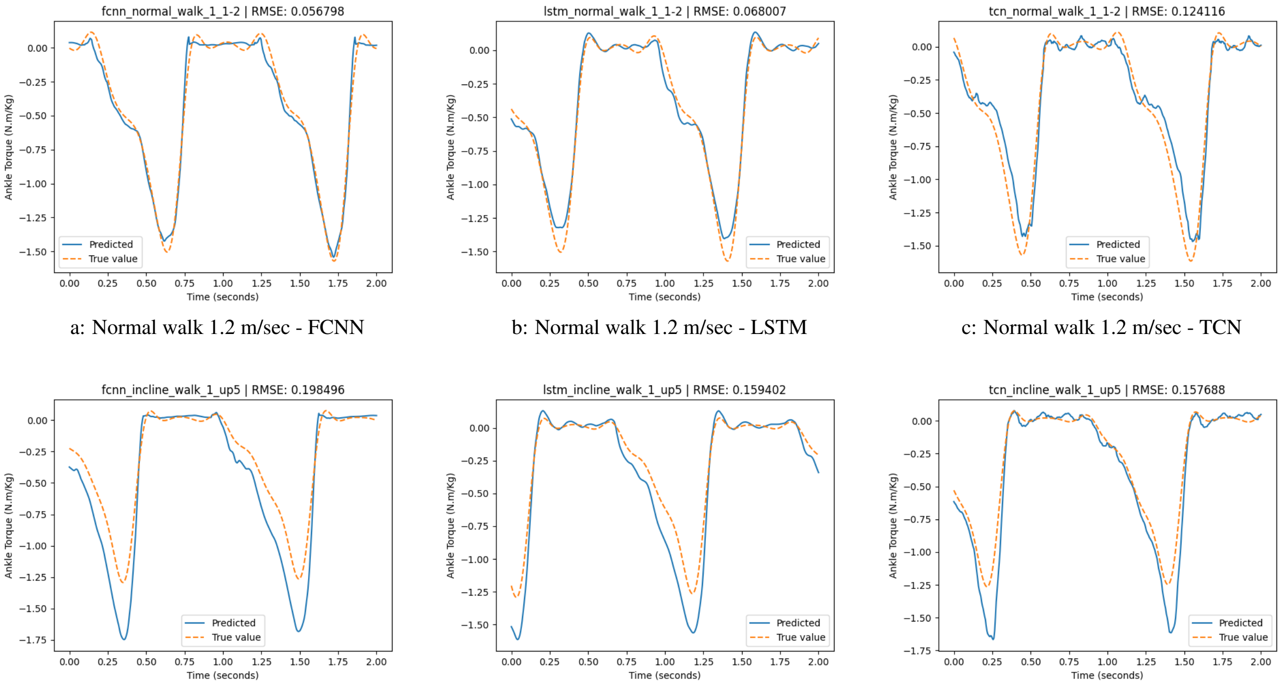
In evaluating the models on the testing tasks, it was observed that the TCN model outperformed both LSTM and FCNN in 14 out of 21 tasks. However, the performance of LSTM and FCNN was comparable for most tasks. It is worth noting that the relatively poor performance of LSTMs could be attributed to the effect of discontinuity in data on the LSTM model hidden states. Unlike the TCN model, which relies on dilated convolutions to capture long-term dependencies in the data, LSTMs use recurrent connections to propagate information across time. The performance of LSTMs can be affected by abrupt changes or discontinuities in the data, which can result in unstable or divergent hidden states.

A notable observation can be drawn from Figure 4, which shows that the simpler FCNN model outperformed both the TCN and LSTM models by a significant margin for variations of normal walking and front-back jumping tasks. This could be attributed to the possibility of overfitting of the FCNN model on the training dataset, which mainly consisted of running and jumping tasks with similar input-output relationships. On the other hand, for more varied tasks such as obstacle walk and sit-to-stand, which rely more on the underlying learned physical model, the TCN model performed very well.

In addition to evaluating the performance of each machine learning model, the computational efficiency of the models can be a crucial factor in real-time applications or when dealing with large amounts of data. In this study, the TCN model demonstrated the fastest computation time among the three models evaluated, averaging around 6μs per prediction. The parallel processing of dilated convolutions allowed the TCN to efficiently process long data sequences without needing recurrent connections. On the other hand, the FCNN model had a longer prediction time averaging around 60μs due to more significant number of parameters in the fully connected layers, which require more computations during training. The LSTM model's computation time was the longest, averaging 300μs per prediction, as it is a recurrent neural network that requires sequential processing of data, which can be computationally expensive.

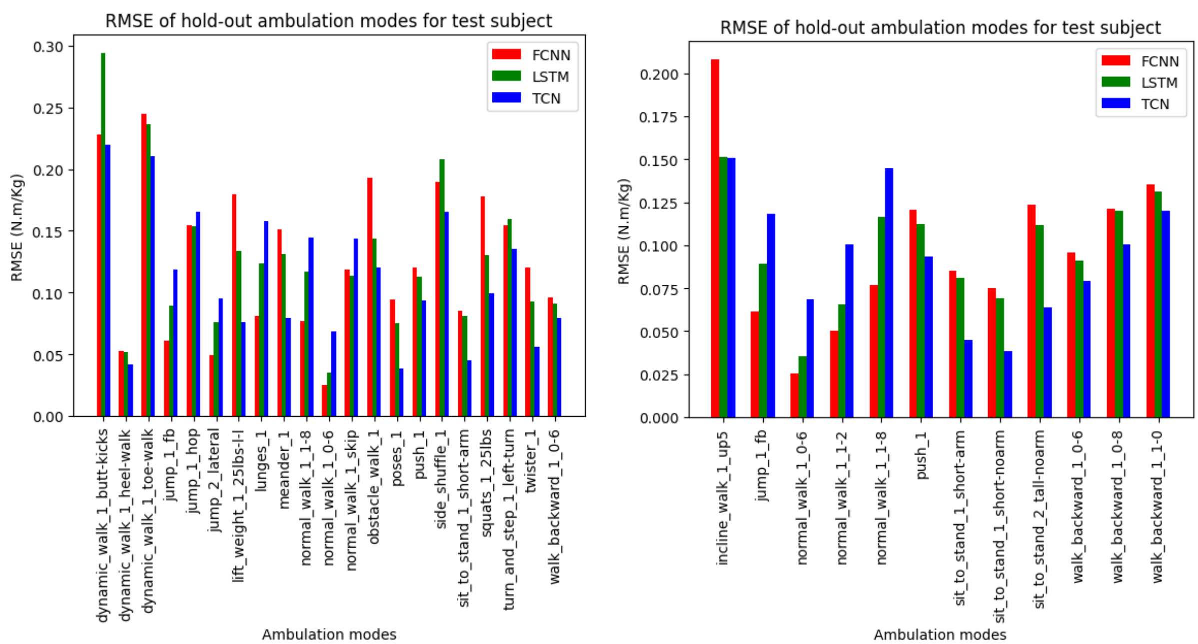
5. Conclusion

Overall, the results demonstrate the effectiveness of using Machine Learning models for estimating ankle joint torque from wearable sensor data. The TCN model show promise for accurately predicting ankle joint torque and can potentially be used as a high-level controller for the ankle



a: Normal walk 1.2 m/sec - FCNN
b: Normal walk 1.2 m/sec - LSTM
c: Normal walk 1.2 m/sec - TCN
d: Incline walk 5 degree - FCNN
e: Incline walk 5 degree - LSTM
f: Incline walk 5 degree - TCN

Figure 3. Predicted Torque curves for normal walking at 1.2m/s (Fig. a, b, c) and ramp walking at a 5-degree incline (Fig. d, e, f) are represented for different ML models. Ground truth for each sample point is also depicted for comparison.



a: Holdout tasks
b: Subset of Holdout tasks relevant to Ankle Exoskeleton

Figure 4. Evaluated RMSE(N.m/Kg) of FCNN, LSTM, and TCN model on test subject for holdout tasks (all in Fig. a and exoskeleton specific in Fig. b)

exoskeleton. Moreover, the computational efficiency of the TCN model makes it suitable for real-time applications or when dealing with large amounts of data.

6. Future Works

Firstly, testing the model performance under varying sensor mounting positions can provide valuable insights into making the model more generalizable for applications in exoskeletons. As sensor mounting positions are not always repeatable, evaluating the impact of such variations can help identify potential challenges and solutions to make the model more robust.

Secondly, incorporating burn-in layers within the LSTM model can help analyze the effect of discontinuity in data on the hidden states of the model. By doing so, the model learns to recover from the bad recurrent initial states [4]. This helps in building a more robust LSTM model, which can be fairly compared with other models.

Lastly, employing a systematic approach for hyperparameter optimization is essential for achieving the best performance for each model. Optimizing hyperparameters allows the model to be fine-tuned to fit the data better, resulting in improved overall performance. Therefore, exploring these areas can help enhance the model's capabilities and pave the way for more accurate and reliable predictions.

7. Acknowledgement

I want to express my sincere gratitude to Prof. Ehsaan Elhamifar and Prof. Max Shepherd for their valuable guidance and support throughout this project. Their mentorship has helped shape my understanding of the subject matter and provided me with the necessary tools and resources to complete this work.

References

[1] Inverse dynamics calculations during gait with restricted ground reaction force information from pressure insoles. *Gait Posture*, 23(2):189–199, 2006. 1

[2] Musculoskeletal model-based inverse dynamic analysis under ambulatory conditions using inertial motion capture. *Medical Engineering Physics*, 65:68–77, 2019. 1

[3] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018. 4

[4] Steven Kapturowski, Georg Ostrovski, John Quan, Rémi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *International Conference on Learning Representations*, 2018. 6

[5] Dean D. Molinaro, Inseung Kang, Jonathan Camargo, Matthew C. Gombolay, and Aaron J. Young. Subject-independent, biological hip moment estimation during multimodal overground ambulation using deep learning. *IEEE Transactions on Medical Robotics and Bionics*, 4(1):219–229, 2022. 1, 2, 3

[6] Dean D. Molinaro, Inseung Kang, Jonathan Camargo, and Aaron J. Young. Biological hip torque estimation using a robotic hip exoskeleton. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 791–796, 2020. 1

[7] Bernd J. Stetter, Frieder C. Krafft, Steffen Ringhof, Thorsten Stein, and Stefan Sell. A machine learning and wearable sensor based approach to estimate external knee flexion and adduction moments during various locomotion tasks. *Frontiers in Bioengineering and Biotechnology*, 8, 2020. 1, 2, 3