Machine Learning Driven Individualized Gait Rehabilitation: Classification, Prediction, and Mechanism Design

Article *in* Journal of Engineering and Science in Medical Diagnostics and Therapy · February 2020

DOI: 10.1115/1.4046321

3 authors:

Amol Loya
Stony Brook University

Shrinath Deshpande
Unity Technologies

Anurag Purwar
Stony Brook University

Machine Learning-Driven Individualized Gait Rehabilitation: Classification, Prediction, and Mechanism Design

Amol Loya
Computer-Aided Design and Innovation Lab,
Department of Mechanical Engineering,
Stony Brook University,
Stony Brook, NY 11794-2300

Shrinath Deshpande
Computer-Aided Design and Innovation Lab,
Department of Mechanical Engineering,
Stony Brook University,
Stony Brook, NY 11794-2300

Anurag Purwar[1]
Computer-Aided Design and Innovation Lab,
Department of Mechanical Engineering,
Stony Brook University,
Stony Brook, NY 11794-2300
e-mail: anurag.purwar@stonybrook.edu

This paper presents a machine learning-based approach toward designing individually targeted rehabilitation devices. This approach consists of a classification model for early detection of a disease, regression, and artificial neural networks (ANNs) models to predict the target rehabilitation gait for a specific individual, and finally a generative approach for the conditional synthesis of single degree-of-freedom linkage mechanisms for gait rehabilitation. Design of mechanisms for human–machine interaction involves numerous subjective criteria and constraints in addition to the motion task. This is particularly important for the rehabilitation devices, where the size, complexity, weight, cost, and ease of use are critical factors. In this paper, we present an end-to-end computational approach for developing a device for individualized gait rehabilitation using machine learning techniques focusing on gait classification, prediction, and specialized device design. These models generate a distribution of linkage mechanisms, which strongly correlate to the distribution of target path variations. This way of formulating the problem results in a large variety of solutions to which subjective criteria can be applied to yield practically useful design concepts that would otherwise not be possible using traditional synthesis methods. [DOI: 10.1115/1.4046321]

Keywords: machine learning, variational auto-encoders, gait classification, gait prediction, path synthesis, mechanism synthesis, mechanism design, motion synthesis, six-bar linkage

1 Motivation

In 2016, 25.7% of the U.S. adults reported some form of disability, while 13.7% of them identified mobility as the most frequent type of the disability. Prevalence of any type of disability was highest among people aged over 45 years old. Half of the individuals presenting difficulty in physical functioning have a gait disability [1]. Common causes for gait disabilities are injuries to the central nervous system, traumatic brain injury, and spinal cord injury causing the loss of motor control. Out of all these injuries, stroke is the most prevalent, resulting in 795,000 new injuries every year in the U.S. alone and it is estimated that 77–80% of those will lead to motor function impairment [2,3]. These numbers demonstrate the severity of the problem and the need for efficient gait training systems. Majority of the gait training systems are designed for acute and subacute neurological inpatients.

Gait training or gait rehabilitation is the act of learning or relearning how to walk, after sustaining an injury or disability. Gait training can be useful for people with amputation, osteoarthritis, muscular dystrophy, cerebral palsy, stroke, Parkinson's disease (PD), multiple sclerosis, brain and spinal cord injuries, and postsurgery and sports injuries.

The most prevailing and accepted method for gait training is physical therapy. Assistive walkers have long been used to aid in gait rehabilitation. Gait training can also be carried out using parallel bars, treadmills, body support systems, or a combination of these systems. But the downside to these training methods is that the individual undergoing rehabilitation need physiotherapists to guide, supervise, and physically help them with the training. Our motivation is to provide a support system for physiotherapists throughout the gait rehabilitation process.

2 Introduction

Body weight supported treadmill training has become a prominent method in gait rehabilitation due to the major advantage of allowing individuals to start training very early in the recovery process. Body weight supported systems can be used by individuals without adequate motor control or sufficient strength to fully bear their body weight. This type of training has many functional benefits, but the labor costs are considerable.

Another approach is to use powered lower limb orthoses for gait training. Exoskeleton robots or robot-assisted gait training systems, mainly constructed by mechatronic components, have emerged in recent years to assist disabled individuals who survived from stroke or other gait disability and do not have enough strength to move their limbs freely on their own. Wolff et al. [4] present the clinical issues with the exoskeleton technology but discusses the potentials of this technology in combination with physiotherapy for gait rehabilitation. The robotic systems that have made significant development in gait rehabilitation are Lokomat [5], MIT-skywalker [6], and KineAssist [7]. These Robotic devices can be classified into two types: exoskeletons [8–10] and end-effector robots [11–13]. However, most of these devices and systems are expensive, bulky, tedious, and difficult to put on. Díaz

[1]Corresponding author.

Contributed by the Applied Mechanics Division Technical Committee on Dynamics & Control of Structures & Systems (AMD-DCSS) of ASME for publication in the Journal of Engineering and Science in Medical Diagnostics and Therapy. Manuscript received October 11, 2019; final manuscript received February 3, 2020; published online March 11, 2020. Assoc. Editor: Ping Zhao.

et al. [14] have provided a comprehensive review of robotic systems and their challenges for lower limb rehabilitation.

In order to reduce the need for complex control logic, researchers have created single degree-of-freedom mechanisms for gait rehabilitation tasks. This is done as follows: The mechanism provides the natural motion for the limbs while the person is adequately supported during ambulation. During ambulation, a person will be putting foot on the ground and experiences the reaction forces. If the person is not able to ambulate, then adequate active or passive resistance can be added to the mechanism so as to force the individual to apply forces as they move their limbs. Shao et al. [15] have presented a cam-linkage mechanism for gait rehabilitation. Tsuge and McCarthy [16] have presented an adjustable six-bar linkage for natural walking movement rehabilitation. The above methods depend upon classical precision point-based approaches like Deshpande and Purwar [17], and Ge et al. [18] to solve the synthesis problems. The precision point approach attempts to find the solutions that best fit the task path; however, this discards large proportions of the solution space, which comprises of linkages with a suboptimal path, but more desirable linkage parameters. Most of the times, optimal numerical fitting of precision points is less important compared to the suitability of fixed pivot regions or proportionate link ratios; for example, a device that implements a single degree-of-freedom mechanism to execute the given motion path may be required to have a smaller frame, fixed and moving pivots within the envelope of the frame, link ratios within a threshold, so as to create a compact and versatile device. To incorporate these aspects for linkage design, we present a machine learning approach for kinematic synthesis, which comprises of a deep generative model that learns the conditional distribution of linkage parameters to their coupler trajectories. Since the method does not use the precision position approach, it finds many suboptimal solutions with other desirable properties like suitable pivot locations and proportionate link ratios. This significantly improves the variety of solutions obtained, which gives more freedom to accommodate other design factors. Moreover, this kinematic synthesis approach can be applied to a linkage database of various topologies to generate linkages for a given task, which makes it scalable to synthesize higher order linkage mechanisms.

The overview of our approach is depicted in Fig. 1. Next, we discuss each of the components shown in this figure.

The initial component of the presented approach is the gait classifier module, which analyzes the current gait of the individual on which various further actions are dependent. Gait analysis is a method for identifying biomechanical abnormalities in the gait cycle. Gait analysis is generally carried out by a medical professional by observing the gait cycle, collecting responses from a questionnaire, and considering the medical history of the individual subject. Observational gait analysis is an acquired skill that requires a lot of training, practice, repetition, and even a professional medical degree for one to detect physical abnormalities. Even after all this, there are chances that the abnormality detected by the clinicians may not be accurate as the methods used for detection are subjective.

Gait information can provide valuable information on the health of the individual subject. Clinical research has identified clear links between human gait characteristics and different medical conditions. In an everyday environment, the state of health or even the age of a person is recognizable by their walk. For example, people generally walk slower as they get older and may start to shuffle; those who have suffered a stroke may drag one of their legs and people with an injured knee or hip joint may walk with an asymmetry [19]. However, in the current medical diagnosis, gait information is rarely used as an input for objective gait analysis or gait classification. This is due to two main reasons: (1) a human can intuitively observe an individual's gait and perform a diagnosis on the gait pattern, and (2) quantifying an individual's gait is difficult.

Tahir and Manap [20] presented an artificial neural network (ANN) and a support vector machine (SVM) model to classify

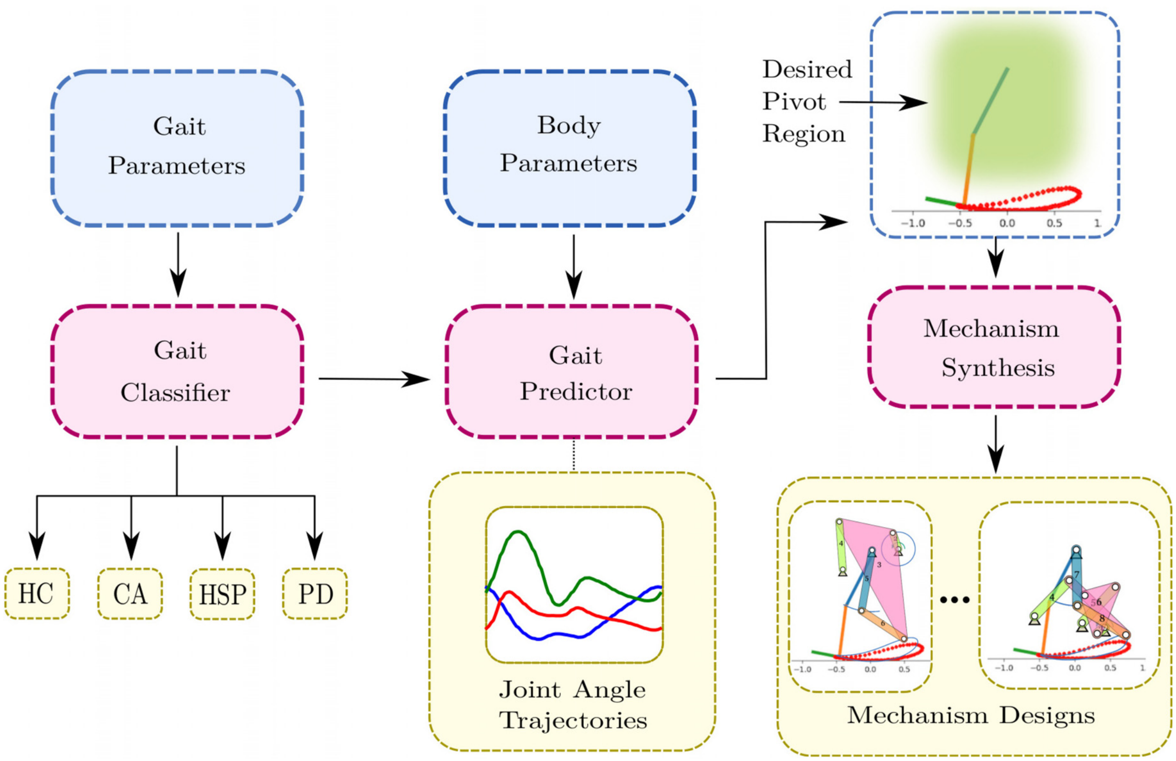

Fig. 1 Gait parameters are used to foretell the gait condition. Individual-specific parameters are used to predict the ankle path necessary for gait rehabilitation. Desired region for fixed pivot is indicated by the shaded region near the leg. The path traced by ankle is used to generate a conditional distribution of four-bar and six-bar linkages having similar paths. Finally, a 2 R link representing thigh and knee are attached to support the entire lower limb.

Parkinson's disease gait against healthy subjects using spatio-temporal, kinematic, and kinetic features of the subject. Yang et al. [21] have classified three types of neurodegenerative diseases from temporal variables obtained using wearable motion sensors. Serrao et al. [22] have proposed a cluster analysis method that can distinguish individuals with different neurological diseases from time–distance parameters and lower limb joint kinematics. All of these methods demonstrate significantly good results, so building on these classification methods, we develop different machine learning driven classifiers to test on the new dataset of gait patterns in neurological diseases [23].

We present here gait classification models, which can be used in clinical settings to assist the medical professional for disease identification. This paper proposes an automatic approach to detect early anomalies in the gait cycle of subjects and also a method to quantify and learn patterns in human gaits. Gait classification is achieved with the general body and gait parameters, which give significant information about the gait. These parameters can be computed with a gait trial by the subject, thus making our gait classification models clinically feasible to complement the medical professionals in the diagnosis of the disease.

Next step is to predict the most probable joint angle trajectory for the specific individual in order to obtain the suitable mechanism for gait rehabilitation. Task specificity is one of the major principles of gait rehabilitation, which suggests that it is crucial to induce a gait pattern that resembles the natural gait pattern of the specifically targeted subject during gait rehabilitation. Banala et al. [10] have presented a robot-assisted gait training in which they use an average healthy gait trajectory as the template for gait rehabilitation. When an individual deviates from this template, a force is provided on the lower limbs through the exoskeleton, which is proportional to the deviation. The proposed gait prediction provides a statistically better template, which considers body parameters than the global average of healthy gaits. Ren et al. [24] implement an inverse dynamics multisegment model of the body and combines it with optimization techniques to simulate normal walking. It derives the segmental motions and ground forces, but with only three inputs, viz., walking velocity, cycle period, and double stance duration, while fixing some of the variables, like gait cycle duration and stride length. Previous machine learning methods have analyzed correlations between gait features (e.g., stride length, walking speed, gait event detection, etc.) and body parameters (e.g., age, weight, height) [25,26]. Aertbelien and De Schutter [27] introduced a gait prediction module to predict the future evolution of gait trajectories, but require the initial gait trajectory as input to make the predictions. Luu et al. [28] proposed a method with an ANN to predict the gait trajectory from gait parameters, such as stride length, cadence, and walking speed. These methods require gait parameters for predicting the trajectories, which may be difficult to find for an individual with severe gait disability.

Yun et al. [29] present a novel statistical method for prediction of gait kinematics with Gaussian process regression. Building on this work, we develop different machine learning techniques to predict the gait kinematics from the body parameters. The approach presented in this paper predicts the walking gait with the knowledge of age, weight, height, gender, and body parameters. Our model presented in this paper is individual-focused and gives personalized prediction results, as it takes into account the individual's body parameters.

Finally, the third module is mechanism synthesis. The goal here is to obtain a variety of mechanism designs that would give the desired gait path. We demonstrate generation of a multitude of single degree-of-freedom mechanism design concepts for the calculated gait profile of a disabled individual.

To achieve this, we use a probabilistic model, called conditional-variational auto-encoders (C-VAE), which allows computing mechanisms similar to the training data. To the best of our knowledge, this is the first attempt at connecting three critical pieces of the gait rehabilitation puzzle: classification, prediction, and device design in the common language of machine learning models. The focus of this paper is on the mechanical design of the device and not the design of an experiment of clinical feasibility.

The organization of the paper is as follows: Sec. 3 presents the techniques used for gait classification along with the classification results on the test dataset. Target gait prediction for an individual based on data-driven techniques is presented in Sec. 4. Finally, Sec. 5 presents the generative approach toward the conditional synthesis of linkages for gait rehabilitation. Section 6 concludes the paper.

3 Gait Classification

The first problem is that of gait classification. As gait information can be used to provide vital information about the individuals' health, we propose different models to extract this valuable information and classify both healthy and disabled individuals from their gait data. This is to assess the effectiveness of models against this type of dataset, which has been largely unexplored. The goal of gait classification is to assist the medical professionals by providing a supplementing tool in the diagnosis of the disease.

3.1 Description of the Dataset. The dataset [23] on gait patterns in degenerative neurological diseases was used to test our framework for gait classification. The dataset provides the gait parameters and lower limb joint kinematics of 142 individuals, which included individuals with three different types of primary degenerative neurological diseases, namely, 19 individuals with cerebellar ataxia (CA), 26 individuals with hereditary spastic paraparesis (HSP), 32 individuals with PD, and 65 individuals as healthy control (HC) group. An opto-electronic motion analysis system was used to measure time–distance parameters and lower limb joint kinematics.

General gait parameters are considered as inputs, which are easy to acquire without requiring a complex system for its measurements, so that this classification model can be reused with ease in clinical or professional setting to assist the medical personnel. The 22-dimensional input feature vector for our machine learning models consists of anthropometric characteristics, typically gender, age, height, and weight along with the gait spatiotemporal and joint kinematics parameters, such as stance duration (time interval between two consecutive heel strikes of the same lower limb), swing duration (time interval between toe off and the next heel strike of the same lower limb), double support duration (time interval with both feet on the floor), step length, speed, hip range of motion (ROM), knee ROM and ankle ROM for each leg and cadence (number of steps per minute), and step width for a gait cycle.

All 142 subjects with their respective 22-body and gait parameters form the input for our model. Given the input gait parameters, the problem is that of multiclass classification predicting the output class as CA, HSP, PD, or HC.

3.2 Preprocessing and Model Setup. As the number of samples for each class are unevenly distributed (19 for CA, 26 for HSP, 32 for PD, and 65 for HC), there is an imbalance in the number of samples for each class. This is called the *class imbalance* problem and it occurs when the class distributions are highly imbalanced. This may result in a biased model, which will give poor generalization. Therefore, we use the resampling technique and try to make the number of samples for each class roughly the same by repeating the samples for classes with minimum frequencies.

After applying the resampling technique, the input gait parameter matrix is enlarged to the dimension (238×22), where each row represents the individual subject and each column represents a gait parameter. The input features are preprocessed and normalized with L^2 norm to get a uniform scale of the data so that our model implicitly weights all features equally in their representation. The output Y for the problem is a scalar label of 238 samples and is therefore of size (238×1).

After normalizing the data, we use principal component analysis [30], which uses an orthogonal transformation to convert the data of possibly correlated variables into a set of values of linearly uncorrelated variables called Principal Components. Principal component analysis is a technique that not only reduces the dimension of the data and improves the speed of convergence for learning algorithm but also helps to enhance the quality of the results. We use the top eight principal components for the classification task for each model, as they cover the maximum variance of 97.17% of the original data. The dimension of the input data is now reduced to (238×8) from (238×22) and will act as the transformed input for the model.

The dataset is then split into training and test set. The model is trained on the training set and the performance of the trained model is evaluated on the test set. Grid search method is used for tuning the hyperparameters for a model. Hyperparameters are nontrainable variables that need to be tweaked for achieving optimum performance for the model. We use cross-validation (CV) technique called the k-fold CV [31] with tenfolds on the training set to estimate the performance of our models. All the possible combinations of the hyperparameters are tried and the model with the hyperparameters that achieves the maximum accuracy on the cross-validation set is selected, and we get the accuracy on the test set with the selected model. Figure 2 presents an overview of the gait classification framework for training the model with the data, and inference for using the trained model to get predictions on unforeseen future samples. In Sec. 3.3, we present three different classification models for predicting disease types and the results of the computational experiments.

3.3 Gait Classification Models

3.3.1 Support Vector Machines. The standard soft margin SVM is a nonprobabilistic linear classifier, which attempts to find the best hyperplane that separates all data points of one class from those of the other class. SVM aims at finding the optimal hyperplane that maximizes the margin between linearly separable classes. We use the one-versus-one approach for multiclass classification. The loss function for training the SVM model is given by

$$L = \frac{1}{N}\sum_{i=1}^{n} \max\Big(0, 1 - \widehat{Y}_i(wX_i - b)\Big) + \lambda ||w||^2 \tag{1}$$

where the first term is the hinge loss which minimizes the mispredictions of the class and the second term is the regularization term which maximizes the separating decision boundary between the classes. N is the number of training samples, and λ [32] is the regularization hyperparameter that gives a tradeoff between correct classification of training examples against maximization of the decision function's margin. $\widehat{Y}_i$ is the actual output and $(wX_i - b)$ is the predicted output by the SVM model. The objective is to learn the parameters w and b such that the loss is minimized. If the original data are not separable by linear decision boundary, the *kernel* trick is used to transform the data into higher dimension to make it linearly separable. The different kernel functions that we try using grid search are linear, polynomial, and radial basis function [33].

After training the SVM classifier model on different hyperparameters using grid search, the best set of hyperparameters obtained were λ as 8 and a kernel as radial basis function. The SVM model with the best hyperparameter set gives an accuracy of 90.53% on cross-validation set and 95.83% on the test set.

3.3.2 Artificial Neural Networks. Artificial neural networks [34] have the ability to learn and model nonlinear and complex relationships, which is important in real life (like in our case), because many of the relationships between inputs and outputs are nonlinear and complex as well.

A standard feed-forward neural network is implemented for pattern recognition. The training cases are used to train the neural net using back-propagation, whereas validation cases are used to test the reliability of the network. The network is trained using

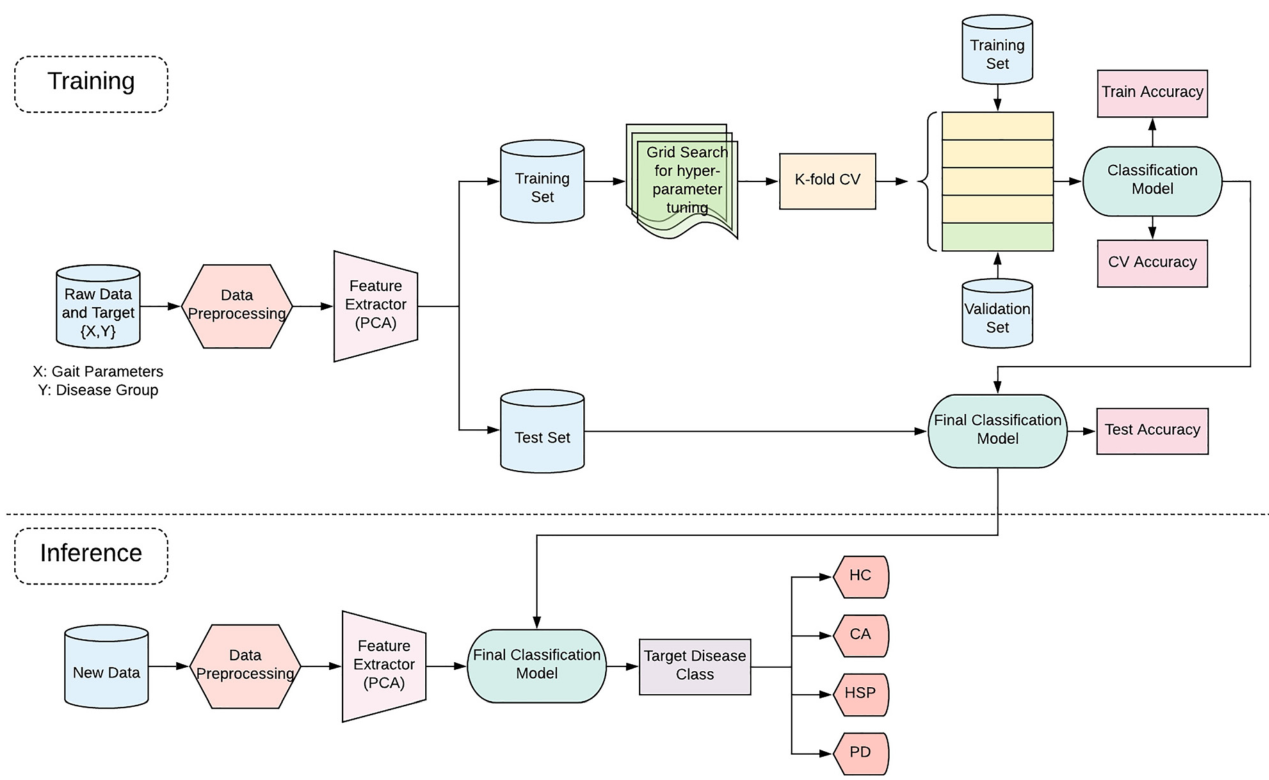

Fig. 2 Gait classification framework

cross-entropy loss function and *Adam* optimizer [35]. The loss function is given by

$$E = -\frac{1}{N}\sum_{i=1}^{n} \widehat{Y}_i \log(Y_i) \quad (2)$$

where $\widehat{Y}_i$ is the ground truth, Y_i is the predicted value by the artificial neural network, and N is the number of training samples. This equation gives the error in the predictions. The goal of the artificial neural network model is to learn the parameters for the neural network (in turn the output, Y) to be as close to the ground truth, $\widehat{Y}_i$, as possible, which is achieved by minimizing the loss function. The hyperparameters to be tuned for the ANN model are the number of layers and number of hidden neurons in each layer, the maximum number of iterations for training the network, type of activation function used to introduce nonlinearity in order to learn complex function (ReLU [36], tanh, logistic) after each layer except the output layer where we use sigmoid activation function.

After training the ANN model on different combinations of the hyperparameters using grid search, the model achieves the best accuracy on cross-validation set with the following hyperparameters: single-layer ANN with 16 neurons, ReLU activation function, and maximum iterations of 250. This ANN classifier model gives a classification accuracy of 94.21% on the cross-validation set and a test accuracy of 95.83%.

3.3.3 Random Forests. Random forests (RFs) are ensemble algorithm, which combines more than one algorithm of decision trees from a randomly selected subset of the training set for classifying the sample. A decision tree is a graph that uses a branching method to illustrate every possible outcome of a decision. The hyperparameters that need to be tweaked using grid search method are the number of trees in the random forests and the maximum depth of the trees.

Random forests classifier, with 40 decision trees and maximum depth of nine, achieves an accuracy of 94.21% on the cross-validation set and test accuracy of 100%.

3.4 Gait Classification Results. Gait classification task was carried out using aforementioned classification models and experimental investigations showed that the random forests classifier outperformed all the other learning models. The accuracy achieved by the random forests model on the unforeseen data was 100%. Even with such a small and complex dataset, we achieved a perfect classification accuracy with the random forests classifier. Table 1 encompasses the CV accuracies using k-fold cross-validation and the accuracies on the test set, for all the gait classification models.

The classification model from their gait parameters predicts if the individual is healthy or is showing signs of inclination to any of the other three classes of disease. This can help in early detection of the disease and help the individual take appropriate measures for gait correction or other health corrections. Also, if our model predicts that the individual is showing an inclination toward a certain disease, they can take therapeutic measures with the proposed hypothesis in Secs. 4 and 5. This classification can also complement with the diagnosis performed by a medical professional.

The classification model also can be used to monitor the health condition of the individual under gait rehabilitation. The individual, who has already been classified for a specific disease by our model, can regularly update their gait parameters during gait rehabilitation. Once their gait parameters are similar to that of a healthy individual, the classification model will output the individual as healthy. This approach would at the least provide some initial data points for a clinician to make an informed decision and provide additional recommendations. These gait classification methods are in no way to replace disease identification by the medical professionals, but to aid them in confirming the onset of the disease.

Table 1 Accuracies for the gait classification models

Method	SVM	ANN	RF
CV accuracy	90.53%	94.21%	94.21%
Test accuracy	95.83%	95.83%	100%

4 Gait Prediction

The goal of gait prediction is to be able to predict the gait trajectories for an individual given their body parameters. In order to be able to do that, we first map the healthy individuals' gait trajectories to their body attributes.

4.1 Description of the Dataset. KIST human gait pattern dataset provided in Ref. [29] was to test our gait prediction framework. Gait trials were carried out on 108 healthy individuals on treadmill with no known history of pathological condition. The body parameters, such as age, weight, height, gender, calf length, thigh length, and other eight lower limb parameters, were collected for the individuals prior to the gait trials. The gait trials captured the sagittal lower limb joint angles for hip, knee, and ankle varying with respect to time. The joint angle trajectories are considered for a single gait cycle and time normalized for 80 time-steps, for all the individuals.

4.2 Preprocessing and Model Setup. We would like to predict joint angle trajectories for a gait cycle for a specific individual given the body parameters for that individual. This is a multivariate time-series prediction problem. For the prediction of joint motions, the training input vector, X and the corresponding training output, Y are defined as follows:

$$X = \begin{bmatrix} X_1^T \\ \vdots \\ X_n^T \\ \vdots \\ X_N^T \end{bmatrix}, \quad Y = \begin{bmatrix} h_{1\times 1} & k_{1\times 1} & a_{1\times 1} \\ \vdots & \vdots & \vdots \\ h_{n\times t} & k_{n\times t} & a_{n\times t} \\ \vdots & \vdots & \vdots \\ h_{N\times T} & k_{N\times T} & a_{N\times T} \end{bmatrix}$$

where N is the total number of subjects, t denotes the time index at which the joint angle is measured, and T $(= 80)$ is the last index of the gait cycle. X_n is an input vector denoting the body parameters for the nth subject, and Y are the corresponding output of the three joint angle predictions at all 80 time-steps for hip extension ($h_{n\times t}$), knee flexion ($k_{n\times t}$), and ankle plantar flexion ($a_{n\times t}$).

The input X is a matrix of dimension (108×13) where the rows denotes the number of subjects and the columns denotes their respective body parameters. The output, Y of dimension $(108 \times 80 \times 3)$ or (108×240) after reshaping, indicates the three joint angles for 108 individuals over a gait cycle. The input is preprocessed and normalized to have zero mean and unit standard deviation before feeding it to our machine learning models. After preprocessing, the data are split into training and testing set. The model is trained on the training set and the performance of the trained model is evaluated on the test set. We again use k-fold cross-validation on the training set with $k = 10$, to assess the predictive performance of the models accurately and grid search method for tuning the hyperparameter for the models.

We developed various automatic gait prediction models, and compared their results to find out which model gives the best performance. We use the standard root-mean-square error (RMSE) as the performance evaluation parameter for the regression models, which is a quadratic scoring rule that also measures the average magnitude of the error. It is given by

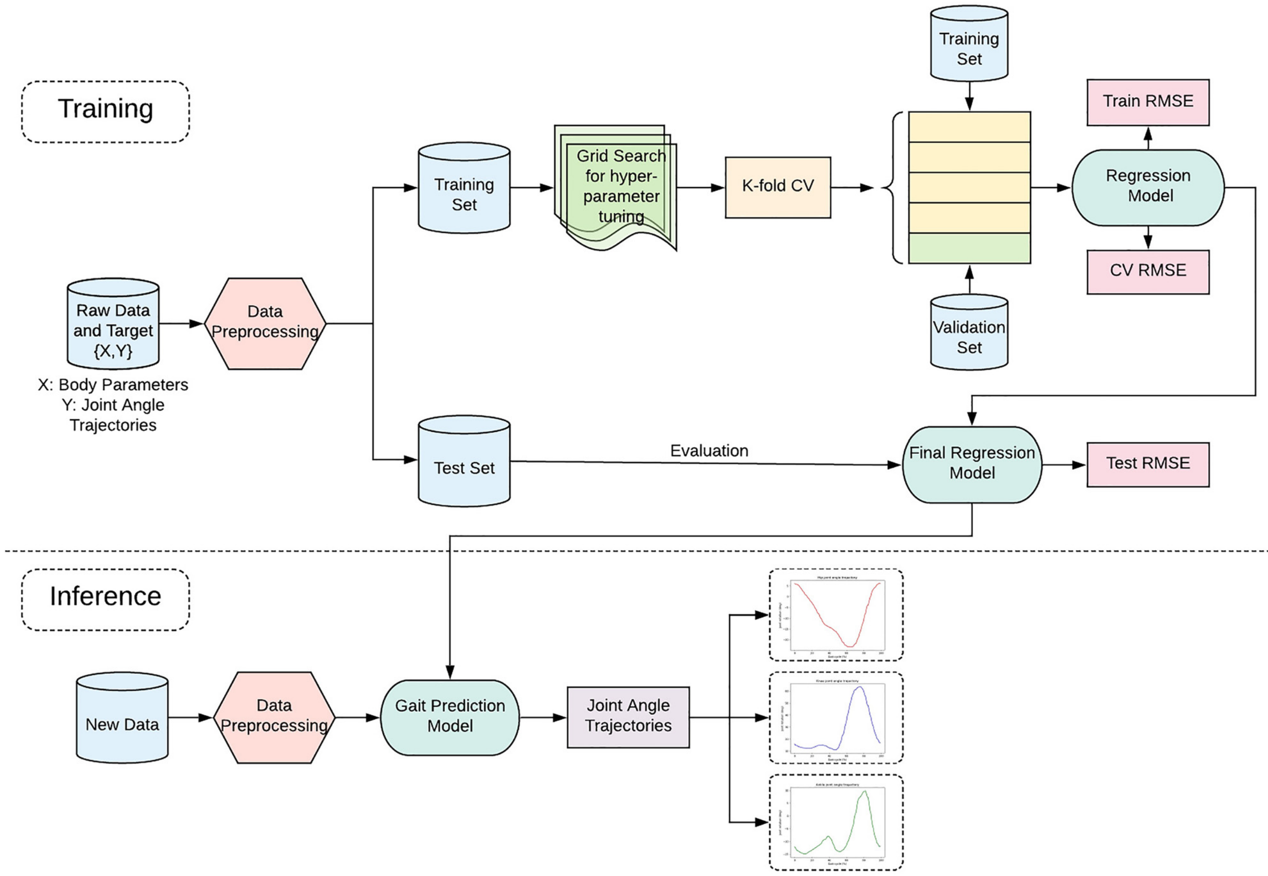

Fig. 3 Gait prediction framework

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}\left(\widehat{Y}_i - Y_i\right)^2} \tag{3}$$

where $\widehat{Y}_i$ is the actual output, Y_i is the predicted output, and N is the number of samples. Figure 3 presents an overview of the gait prediction framework for training the model with the data, and inference for using the trained model to get predictions on the unforeseen future samples. We experiment with three regression models and compare their results, as presented further in this section.

4.3 Gait Prediction Models

4.3.1 Lasso Regression. Lasso regression is an extension built on the linear regression model with a regularization function. Lasso regression is a modification to linear regression with absolute value for the regularization term. It makes a huge impact on the tradeoff between under-fitting and over-fitting [37]. Lasso method overcomes the disadvantage of linear regression by not only punishing high values of the coefficients θ but also setting them to zero if they are not relevant. This leads to lasso regression to achieve extremely good results. The loss function of lasso regression is of the form

$$L = \sum_{i=1}^{n}\left[\widehat{Y}_i - \sum_{j=0}^{m}\theta_j X_{ij}\right]^2 + \alpha\sum_{j=0}^{m}|\theta_j| \tag{4}$$

where $\widehat{Y}_i$ are the actual outputs and $\theta_j X_{ij}$ are the predicted outputs from the lasso regression model; θ's are the weights to be learned, and α is the regularization hyperparameter, which penalizes the large weight values. The objective of training the lasso regression model is to learn the optimum values of the parameters, θ's that minimize the loss function.

After training the lasso regression model, the root-mean-square error, between the predicted and the actual values for the joint angle trajectories, of 7.2412 is achieved on the cross-validation set with $\alpha = 1.0$. RMSE for prediction on test set is 8.2671. Figure 4 shows the predicted and the actual joint trajectories of hip, knee, and ankle for four arbitrary individuals by our lasso regression model.

4.3.2 Polynomial Regression. Polynomial regression is a form of regression analysis in which the relationship between the independent variable X and the dependent variable Y is modeled as a nth degree polynomial in X. Polynomial regression fits a nonlinear relationship between the values of the input, X and the corresponding output, Y. The hyperparameters for grid search for polynomial regression model are the degree of the polynomial function and the regularizer hyperparameter, α. The loss function for polynomial regression is given by

$$L = \sum_{i=1}^{n}\left[\widehat{Y}_i - \sum_{j=0}^{m}\theta_j X_{ij}\right]^2 + \alpha\sum_{j=0}^{m}|\theta_j| \tag{5}$$

where $\widehat{Y}_i$ are the actual outputs, X_{ij} are now the polynomial features, and $\theta_j X_{ij}$ are the predicted outputs from the polynomial regression model; θ's are the weights to be learned, and α is the regularization hyperparameter, which penalizes the large weight values. The objective of training the polynomial regression model is to find the optimum values of the parameters, θ's that minimize the above loss function.

After training the polynomial regression model, the best hyperparameters, which give the least root-mean-square error, are degree of polynomial as 2 and α as 0.1. The polynomial regression

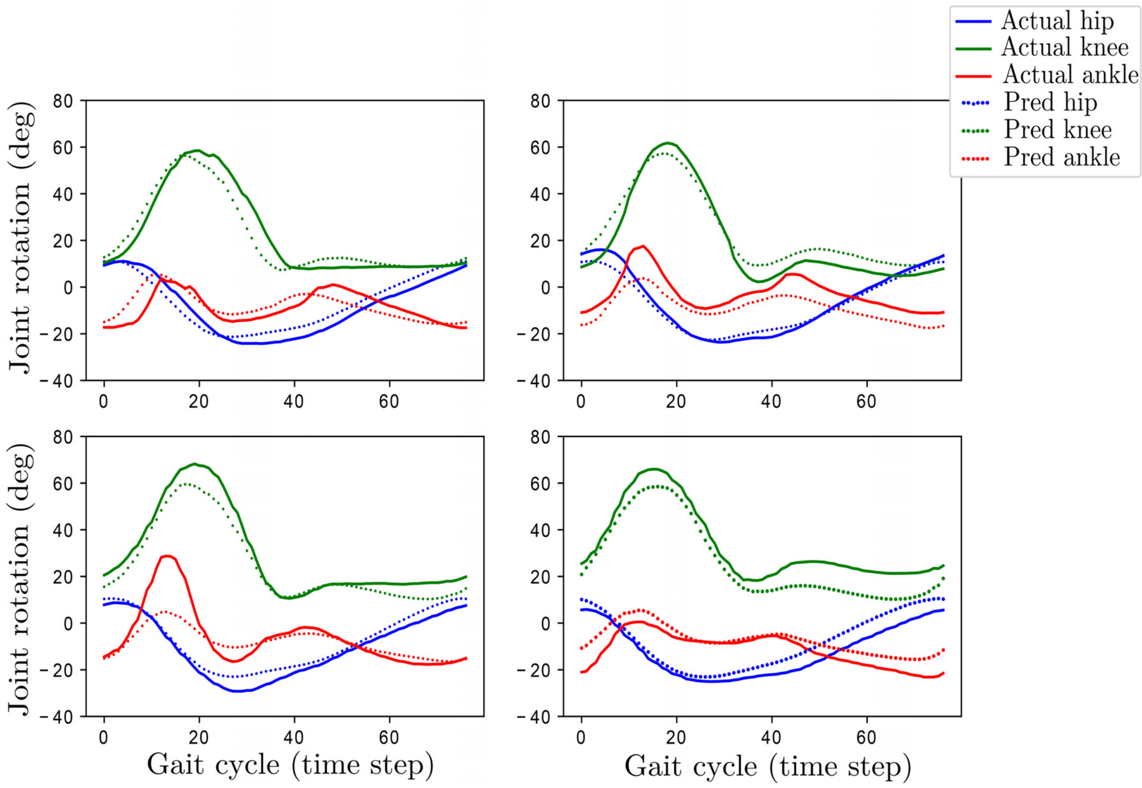

Fig. 4 Predicted angle trajectories of four arbitrary individuals with lasso regression

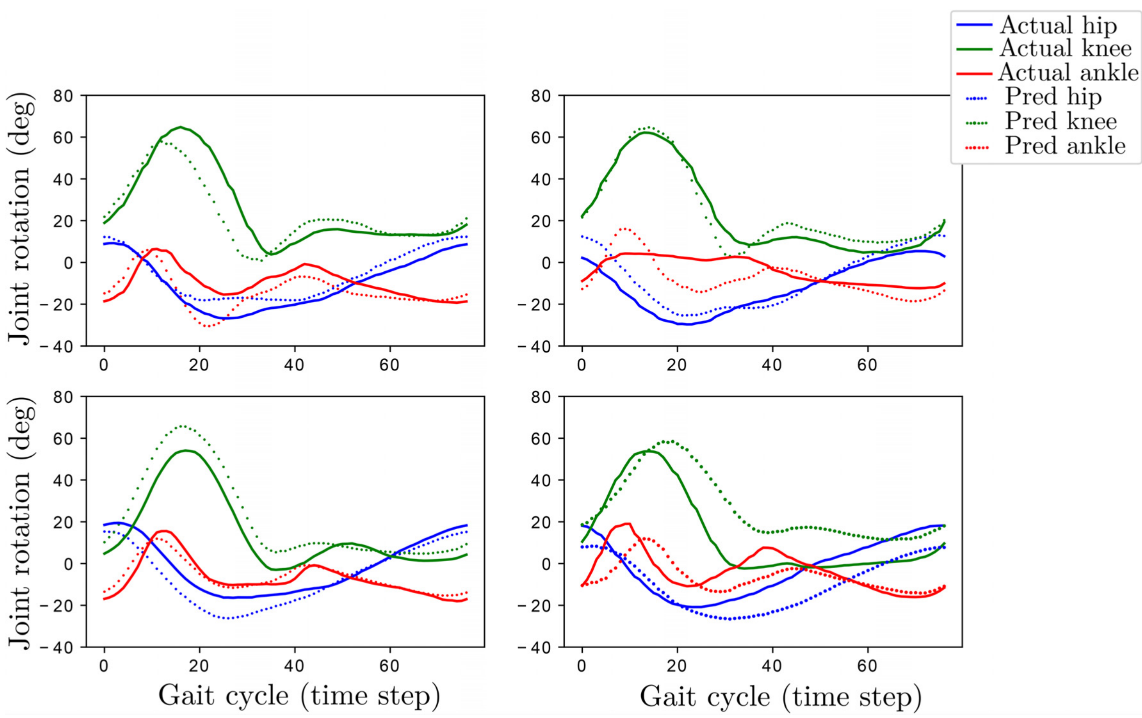

Fig. 5 Predicted angle trajectories of four arbitrary individuals with polynomial regression

model with these hyperparameters gives a RMSE of 10.6978 between the predicted and the actual values of the joint angle trajectories, and a RMSE score of 10.0015 on the test set. Figure 5 shows the gait trajectory predictions of four randomly chosen individuals from the test set with polynomial regression.

4.3.3 Artificial Neural Networks. Artificial neural networks, like in the case of classification, can be used for regression task as well. A standard feed-forward neural network is implemented here to learn the function between the input parameters and the gait angle trajectories. The network is trained using

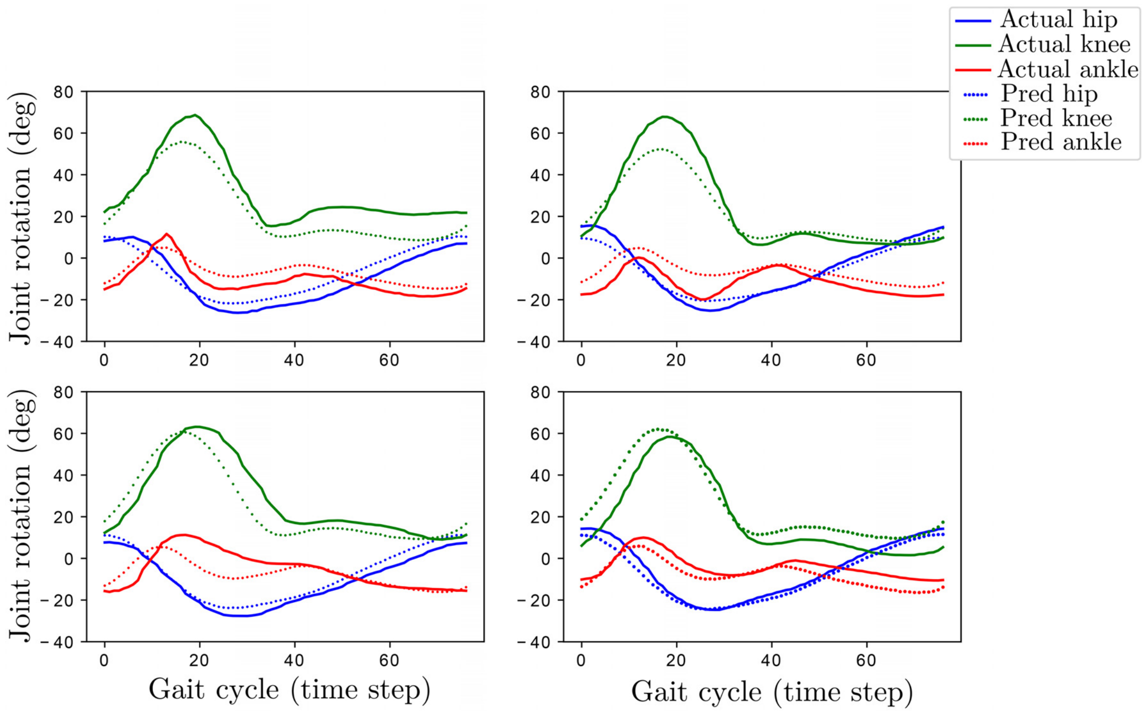

Fig. 6 Predicted angle trajectories of four arbitrary individuals with artificial neural network

back-propagation. The artificial neural network is setup with our preprocessed input, X as the input body parameters and Y as their corresponding joint angle predictions. The hyperparameters for the ANN regressor are same as that in the case for ANN classifier, except that we do not apply any activation at the last layer, as we need the actual values of joint angle predictions. The loss function for training the neural network in this case is the mean squared error, given by

$$E = \frac{1}{N}\sum_{i=1}^{n}\left(\widehat{Y}_i - Y_i\right)^2 \tag{6}$$

where $\widehat{Y}_i$ are the ground truths, Y_i are the predicted values by the artificial neural network model, and N is the number of training samples. This equation gives the error in the predictions. The goal of the artificial neural network model is to learn the parameters θ (in turn the output, Y_i) to be as close to the ground truth, $\widehat{Y}_i$, as possible, which is achieved by minimizing the loss function.

The best set of hyperparameter obtained using grid search, that gives the least RMSE of 7.4125 between the predicted and the actual trajectories, on the cross-validation set are two-layered ANN with 100 neurons each, logistic activation function, and maximum iterations of 500. The RMSE on the test set is 8.5097. The actual and the predicted joint trajectories of hip, knee, and ankle of four individuals is shown in Fig. 6.

4.4 Gait Prediction Results. Gait prediction is a very complex task as it involves prediction of multivariate time series. Still, the models developed here give superior results and show the potential these models have to offer for gait prediction. These models provide a functional relationship between the comprehensive body parameters and gait cycle predictions.

Polynomial regression gives us average results because adding the polynomial features, for the purpose of learning complex function, but that creates more parameters, which are difficult to train. ANN regression model gives good results, but lasso regression model gives us the best predictions along with smooth joint angle trajectories for individual-specific parameters, with least error between the predicted and the actual joint trajectories. Table 2 summarizes the RMSE values of the gait prediction models on the CV and the test set. Table 2 also shows that the least error obtained, between the predicted and the actual trajectories, on the unforeseen test set is by lasso regression model, and thus this model is used for the gait prediction task.

Table 2 RMSE for the gait prediction models

Method	Lasso	Polynomial	ANN
CV RMSE	7.2412	10.6978	7.4125
Test RMSE	8.2671	10.0015	8.5097

The gait prediction models developed return the joint angle trajectories from the body parameters of a specific individual. The benefit of such a model is that given that the individual is following an unhealthy gait pattern, from our classification model, we can define a gait trajectory template for that specific individual from their body parameters alone. Moreover, this model can be utilized for robot-assisted gait training or orthosis systems for determining the customized gait trajectory for a distinct user and train them with their specific trajectory than a target gait cycle. The range of possible applications of this machine learning-based model-driven gait prediction varies from biomechanics analysis to rehabilitation robot control.

Using the predicted joint angle trajectories from the regression model and the leg lengths of the individual subject, we derive the rehabilitation path for individual's ankle using forward kinematics. The rehabilitation path thus includes information about the joint angle motion and the leg lengths of the subject. Figure 7 shows the locus of ankle position trajectories obtained from lasso regression model for the samples in the test set. This provides the information about the rehabilitation task paths that our mechanism needs to generate. With the help of this rehabilitation path, we will carry out mechanism synthesis in Sec. 5.

5 Mechanism Synthesis

Once the gait required for training the individual is obtained, the design process for the device can be initiated. The first step in

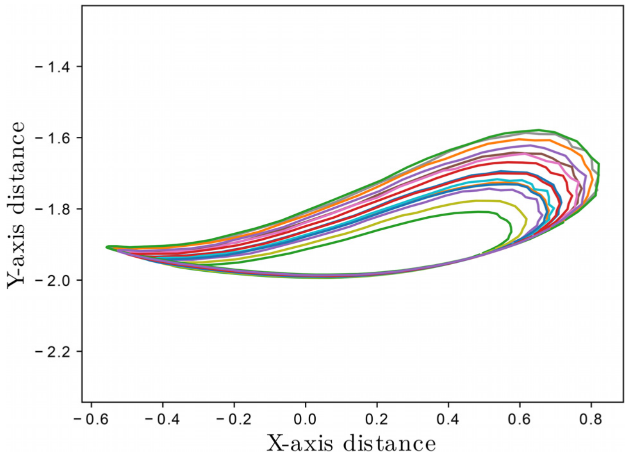

Fig. 7 Predicted ankle trajectories by lasso regressor

the design of the device is to setup the synthesis problem. The open-ended nature of the problem can give rise to different designs, but by marrying a suitable design methodology with the machine learning techniques, it is possible to generate a multitude of solutions, which can satisfy multiple practical constraints. In this paper, we choose to design a one degree-of-freedom device as it removes the need for complex control logic and reduces sensor requirements. This further facilitates highly customized design of devices that address an individual's disability.

5.1 Problem Setup. The task is to find linkage type and dimensions such that the linkage can move the individual's ankle along the prescribed path. Here, the prescribed path is obtained by performing forward kinematics on a kinematic model representing an individual's lower limb. Along with the prescribed path, the fixed and moving pivots of the linkages should lie above the ground throughout the entire simulation. The above constraint on the moving pivots cannot be put into a closed-form equation. On top of this, the mechanism should be compact and should have suitable link proportions. Thus, it is not practical to consider each of such constraints at the first step. So, the approach is to apply these constraints after a set of solutions are found for the path synthesis problem. The solutions which pass these specifications constitute the feasible set of concept solutions.

Therefore, it is desirable to have a path synthesis method, which yields a diverse set of solutions. In order to synthesize diverse planar kinematic linkages, it would be useful to the learn the conditional distribution of linkage parameters from a dataset of unique mechanisms. This section presents our step in this direction by formulating a generative model of linkage parameters.

In this work, we have used C-VAE [38] as our generative modeling framework, which is a directed probabilistic model. The generative model can generate data samples whose probability distribution is similar to that of training data, which are called maximum likelihood samples.

5.2 Conditional Variational Auto-Encoders. Conditional-variational auto-encoders [38] are a neural network architecture that learns to approximate conditional distribution of an observed data X given an observed property Y. Figure 8 shows a general architecture of a C-VAE. In this figure, X and Y are concatenated and fed to Recognition Model as the input. There are two hidden layers h_1 and h_2 in the Recognition Model, while the Generative Model has one hidden layer h_3. In the middle, there is feature space encoded by the variable z, which seeks to capture the salient features of the input data in a compact latent space. Given observed variables X and Y, recognition model computes an approximate probability distribution $q(z|X, Y)$ of the latent variable z as following:

$$\mu, \sigma = Q(X, Y; \theta_e) \tag{7}$$

$$q(z|X) = \mathcal{N}(\mu, \sigma) \tag{8}$$

Here, $\mathcal{N}(\mu, \sigma)$ is a multivariate Gaussian distribution function with mean μ and variance σ. The generative model is a function which is trained to maximize likelihood samples $\widehat{X}$ by taking a sample from $z \in q(z|X, Y)$ and concatenating it with Y. Thus, X is given by

$$\widehat{X} = G(z, Y; \theta_g) \tag{9}$$

In this paper, we assume the prior probability distribution of $p(z)$ as a multivariate Gaussian with mean 0 and unit variance. The training task is to find parameters θ_g, θ_e of $G(z)$ and $Q(X)$, respectively, that maximize an objective, which is a combination of generating maximum likelihood samples and obtaining posterior distribution of z (i.e., $q(z|X, Y)$) close to the true distribution $p(z)$.

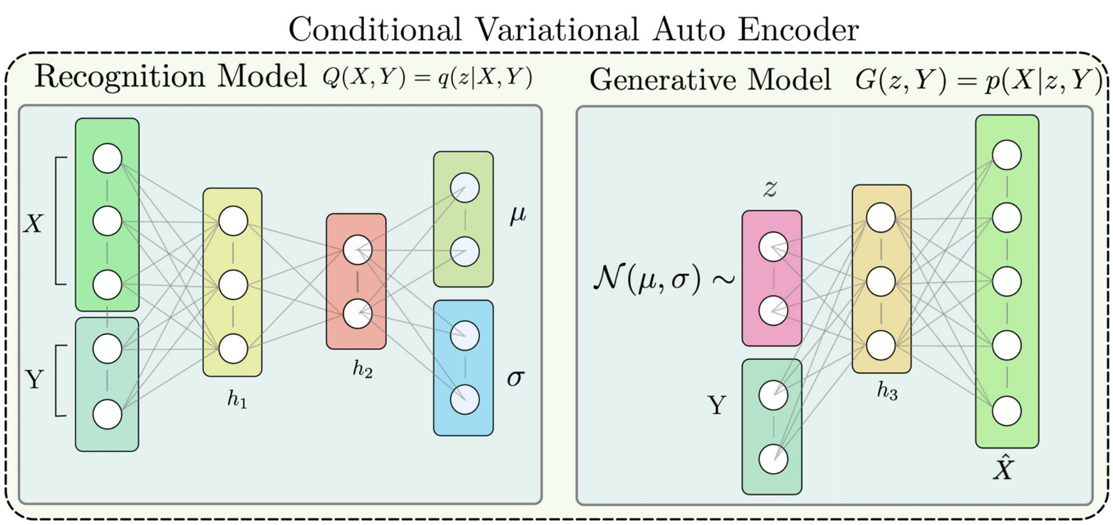

Fig. 8 Recognition model encodes the observed data *X* and observed property *Y* into probabilistic latent coding *z* of dimension much smaller than *X*. In this case, we assume a multivariate Gaussian distribution for *z*. Generative model takes samples from this distribution and combines it with *Y* to generate output $\widehat{X}$.

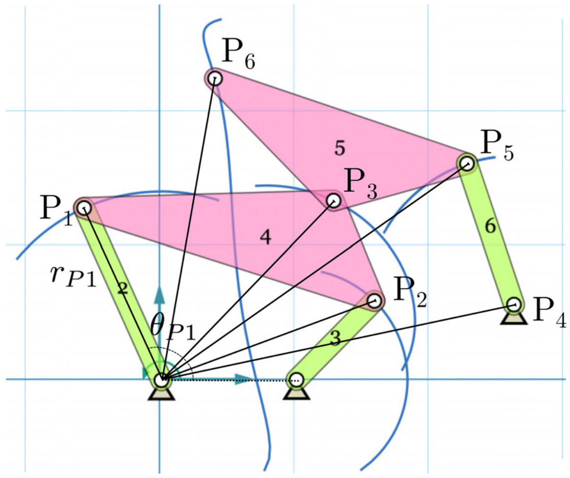

Fig. 9 A Stephenson six-bar linkage with the fixed link of unit magnitude and colinear with *X*-axis. The polar coordinates of points $\{P_i\}_{i=1}^{6}$ stacked together for *m* crank orientations constitute the state representation of the six-bar.

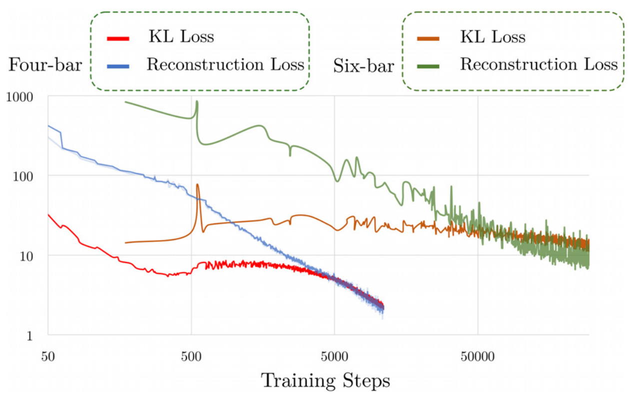

Fig. 10 Reconstruction and KL Divergence losses for C-VAE-FB and C-VAE-SB. The architectural details of these models are given in Table 3.

This is achieved by training the neural network models for minimizing the loss given by

$$L = (\widehat{X} - X)^2 + \sum_{j}^{z_{\text{dim}}} \text{KL}(q_j(z|X, Y)||p(z)) \tag{10}$$

Here, the first term represents reconstruction likelihood and the second term is called Kullback–Leibler divergence (KL divergence) [39], which is the measure of divergence in between the learned distribution $q(z|X, Y)$ and true prior distribution $p(z)$. Note that z_{dim} is the dimension of latent space. KL divergence, when $p(z)$ is a Gaussian distribution with zero mean and unit variance, is given by

$$\text{KL} = \sum_{i}^{z_{\text{dim}}} \sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1 \tag{11}$$

where μ and σ are given by Eq. (8); for further details see Ref. [38]. The reconstruction likelihood imposes a penalty for not

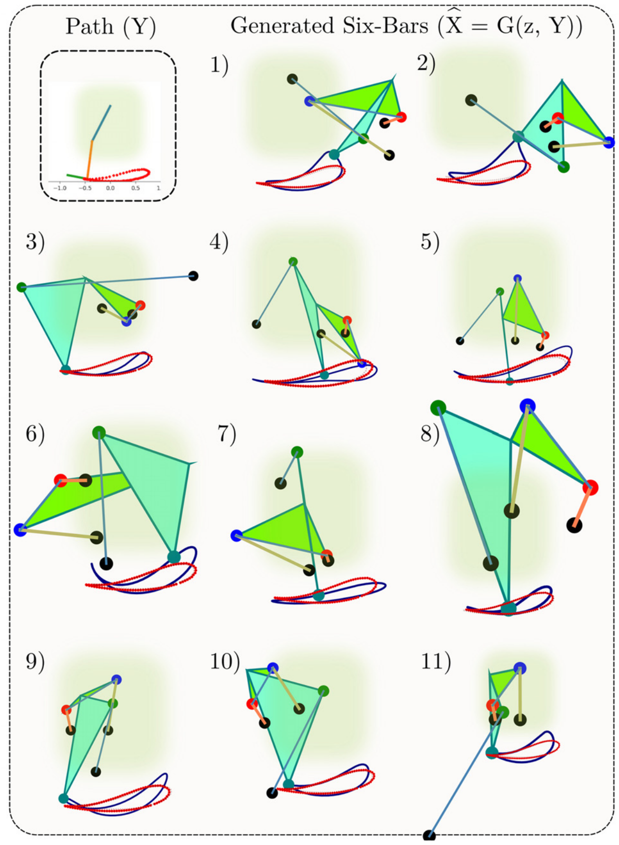

Fig. 11 Stephenson six-bars generated by C-VAE-LS15 (see Table 3) conditioned for coupler curve *Y* and 15 dimensional Gaussian multivariate *z*. The mechanisms having pivots in desired region are selected for the next step.

being able to reconstruct the original data, while the KL divergence term penalizes creation of an excessive number of clusters in the feature space.

Once the entire C-VAE is trained, the generative model can be used separately to perform kinematic synthesis. Parameters of this neural network are learned to map a concatenating vector comprising of latent space and label Y to a reconstruction, which would exhibit similar latent attributes z if passed through the recognition network. The architecture of this model starts with an input layer, which receives the concatenated vector and passes through single or multiple layers of neurons culminating into the original size of the observed data. The middle layers are fully connected layers which upscale the input they receive from the previous layer. We apply Leaky ReLU activation function [40] on the output of each hidden layer. Leaky ReLU is given by

$$\text{ReLU}(x) = \max(\alpha x, x) \tag{12}$$

where α is a small constant, which we take to be 0.001 based on common machine learning practice.

5.3 Training Conditional-Variational Auto-Encoders for Mechanism Synthesis.

Conditional-variational auto-encoders discussed in Sec. 5.2 require a tuple (X, Y) to train, where X is an

Table 3 VAE and C-VAE Model Architectures (FB = four-bar, SB = six-bar)

Observed data (X)	X dimension	Name	Encoder architecture	Latent (z) dim	Decoder architechture	Y	Reconstruction loss	KL loss
FB linkages	100×6	C-VAE-LF10	(300, 100)	10	(100, 300)	$\wp$	1.90	2.24
SB linkages	100×12	C-VAE-LS15	(600, 300)	15	(300, 600)	$\wp$	6.45	12.59

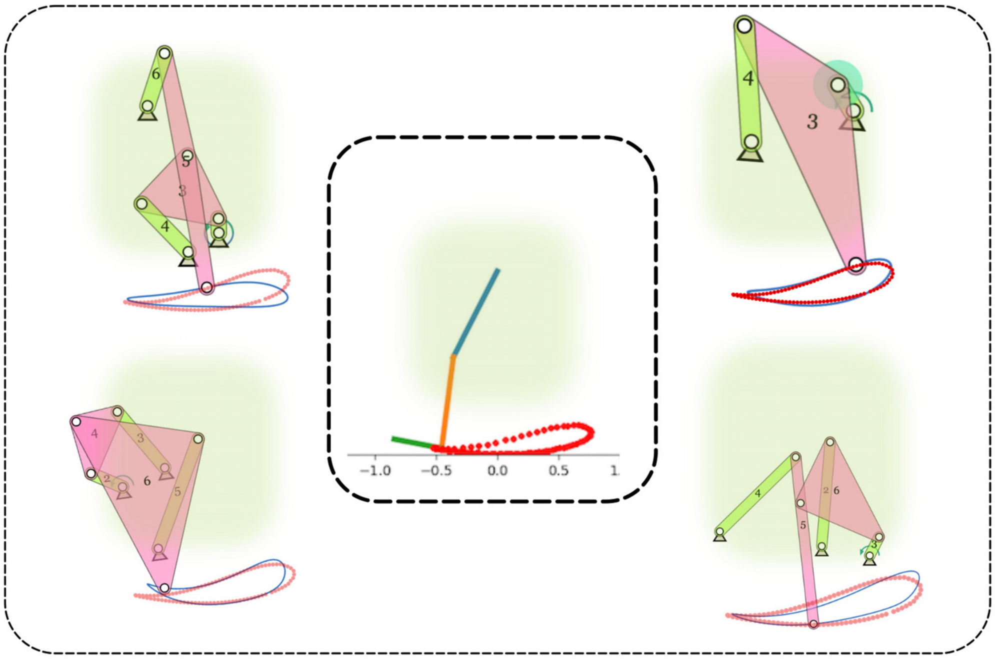

Fig. 12 Collection of feasible six-bar mechanisms with desirable properties. It can be seen that their fixed pivots are in the desired shaded region.

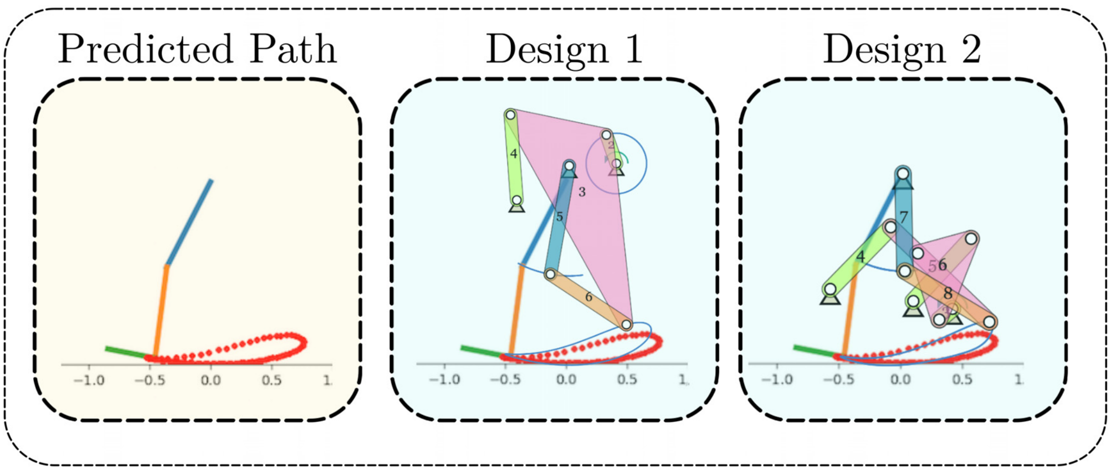

Fig. 13 Final linkage concept solutions

observed variable (in this case the entire linkage parameters) and Y is an observed property or condition (in this case, the coupler curve path). In theory, this formulation should work for any such tuple, which has a strong correlation. The observed variable X is obtained from the linkage parameters such that it should contain the information of the entire simulation of linkage in its current configuration. Also, it is amenable for machine learning to ensure that the X is unaffected by scale, orientation, and translation transformations. To do that, we orient and scale the linkage such that one of its fixed links has magnitude 1 and is parallel to the X-axis. We sample locations of all the points of interest for m crank orientations divided uniformly throughout the possible range. Next, we represent these locations in polar coordinates with origin at fixed pivot corresponding to the crank. Then, these coordinates are stacked together for all of the m orientations. In the case of a Stephenson six-bar linkage, we have six points of interest $P_1, P_2, P_3, P_4, P_5, P_6$ as shown in Fig. 9. Thus, the state tensor X is given by

$$X = \{r_{P1}, \theta_{P1}, r_{P2}, \theta_{P2}, r_{P3}, \theta_{P3}, r_{P4}, \theta_{P4}, r_{P5}, \theta_{P5}, r_{P6}, \theta_{P6}\}_{i=1}^{m} \tag{13}$$

where r_{Pj} and θ_{Pj} are the radial and angular coordinates of point P_j. It should be noted that a tuple $(X = \{r_{P1}, \theta_{P1}, r_{P2}, \theta_{P2}, r_{P3}, \theta_{P3}, r_{P4}, \theta_{P4}, r_{P5}, \theta_{P5}, r_{P6}, \theta_{P6}\}_{i=1}^{m},)$ for any of the orientations can be used to construct the original mechanism back.

X has dimensions $[m, 2 \times \text{PoI}]$, where PoI is the number of points of interest, whereas Y is taken as the corresponding coupler path, which has dimension 2 m. We flatten X and Y to form vectors of dimensions $m \times 2\text{PoI}$ and 2 m, respectively. In order to train C-VAE, we pass a batch of X and Y to the network and compute gradients and losses. The training losses of C-VAE for four-bar and six-bar mechanisms are depicted in Fig. 10. It can be seen in Fig. 10 that C-VAE for four-bar takes lesser time to train with better training accuracy. This observation is supported by the fact the six-bar linkages have higher complexities in the distributions of the parameters. The model architectures and their training results are tabulated in Table 3, where $\wp$ signifies path problem. This table also shows the details of each architecture, such as the number of neurons in each layer, the number of hidden layers, and the losses.

5.4 Conditional Generation of Mechanisms. Conditional-variational auto-encoders for six-bar mechanisms (C-VAE-SB) are trained to map the probability distribution of linkages from the dataset on which it is trained to the shape of their corresponding coupler paths. Now, we use this trained model to generate conceptual designs for the prescribed gait rehabilitation path. C-VAE generates a conditional distribution of linkages for a given path. A hundred samples from this distribution were taken out of which forty of the samples possessed similar coupler curves as the task curve. From those samples, eleven linkages were selected by visual inspection and displayed in Fig. 11 along with the prescribed path Y.

It should be noted that by avoiding to formulate the problem as minimizing the fitting error between the task path and the coupler curve of the mechanisms, we are able to generate a multitude of mechanisms with desired structure. Figure 12 shows a few such feasible mechanisms. It can be seen that some of these mechanisms may have a worse fitting to the task path, but are better suited for this application as their fix pivots lie inside the desired region. Here, the number of solutions obtained is only limited by the sampling of the latent space; however, not every sample would produce a unique or useful linkage.

The concepts obtained in Fig. 12 can be further fine-tuned in detailed design phase. Along with six-bars, C-VAE-FB model (see Table 3) was also used to find out the suitable four-bar linkages. The Figure 13 shows one chosen solution concept along with an additional 2R link, which resembles the thigh and lower leg of the person. It can be seen that fixed pivots and link ratios have appropriate proportions with respect to the lower limb of the person. Using the C-VAE approach, we are able to generate a large sample of mechanisms that can satisfy a variety of highly constrained requirements.

6 Conclusion

In this paper, we have presented an end-to-end approach toward designing individually targeted rehabilitation devices. The approach starts with the classification model that predicts early detection of a disease and provides guidance on the need for gait rehabilitation. The classification model can also be used to monitor the individual under rehabilitation by updating their new gait parameters as the input to the model during training. The next step in the process was training the regression model on the databases containing information about concerned gait cycles of healthy individuals. The regression model is used to predict the rehabilitation path (or motion) suitable for the specific individual. This predicted path (or motion) is taken as the task for the synthesis of mechanisms. Then we apply the deep generative models to obtain conditional distribution of mechanisms for the given task. The generative models are trained to approximate the conditional distribution of a variety of mechanisms given their coupler paths (or motions). This novel approach provides a real-time variational synthesis of mechanisms to generate a set of diverse solutions. Traditional methods usually involve solving high-order polynomial equations. Solving higher-order polynomial equations required significantly intensive calculations of iterative nature, which may take several hours on an average computing machine. Whereas, computing output from a trained neural network is a real-time computation. It can be argued that training neural networks is equally intensive in terms of computation, but at the time of inference, little computing power is needed. Thus, the proposed method uses significantly low computation at the time of synthesis. Since the method is not driven to find the numerically optimal solutions to the gait trajectory, it is not discouraged to find suboptimal solutions with other desirable properties. The method provides a distribution of probable linkages from which any number of linkages can be sampled. However, there is no hard guarantee that generated linkages will resemble the target gait. Further exploration is necessary to compare the performance of the proposed method to traditional synthesis methods.

Designed single degree-of-freedom mechanisms by virtue of their design are comparatively inexpensive for a gait rehabilitation system and also have added benefits of being individually focused, compact, and capable of carrying the gait rehabilitation without the need for complex control. The mechanism obtained can also have a provision to be integrated with assistive walkers and one of the links can be made adjustable as well to get some variation in the trajectory, if desired during the rehabilitation process. In conclusion, we have demonstrated a novel approach leveraging machine learning techniques for gait classification, prediction, and mechanism design for gait rehabilitation, which has the potential to address motion-related challenges for disabled individuals.

Acknowledgment

This work has been financially supported by The National Science Foundation under a research grant to Stony Brook University (A. Purwar and Q. J. Ge, Grant CMMI-1563413). All findings and results presented in this paper are those of the authors and do not represent those of the funding agencies.

Funding Data

- The National Science Foundation (Grant No. CMMI-1563413; Funder ID: 10.13039/100000001).

References

[1] Okoro, C. A., Hollis, N. D., Cyrus, A. C., and Griffin-Blake, S., 2018, "Prevalence of Disabilities and Health Care Access by Disability Status and Type Among Adults-United States, 2016," Morbidity Mortal. Wkly. Rep., **67**(32), pp. 882–887.

[2] Lawrence, E. S., Coshall, C., Dundas, R., Stewart, J., Rudd, A. G., Howard, R., and Wolfe, C. D., 2001, "Estimates of the Prevalence of Acute Stroke Impairments and Disability in a Multiethnic Population," Stroke, **32**(6), pp. 1279–1284.

[3] Sommerfeld, D. K., Eek, E. U.-B., Svensson, A.-K., Holmqvist, L. W., and von Arbin, M. H., 2004, "Spasticity After Stroke: Its Occurrence and Association With Motor Impairments and Activity Limitations," Stroke, **35**(1), pp. 134–139.

[4] Wolff, J., Parker, C., Borisoff, J., Mortenson, W. B., and Mattie, J., 2014, "A Survey of Stakeholder Perspectives on Exoskeleton Technology," J. Neuroeng. Rehabil., **11**(1), p. 169.

[5] Jezernik, S., Colombo, G., Keller, T., Frueh, H., and Morari, M., 2003, "Robotic Orthosis Lokomat: A Rehabilitation and Research Tool," Neuromodulation: Technol. Neural Interface, **6**(2), pp. 108–115.

[6] Susko, T., Swaminathan, K., and Krebs, H. I., 2016, "MIT-Skywalker: A Novel Gait Neurorehabilitation Robot for Stroke and Cerebral Palsy," IEEE Trans. Neural Syst. Rehabil. Eng., **24**(10), pp. 1089–1099.

[7] Peshkin, M., Brown, D. A., Santos-Munné, J. J., Makhlin, A., Lewis, E., Colgate, J. E., Patton, J., and Schwandt, D., 2005, "KineAssist: A Robotic Overground Gait and Balance Training Device," Ninth International Conference on Rehabilitation Robotics (ICORR 2005), Chicago, IL, June 28–July 1, pp. 241–246.

[8] Colombo, G., Joerg, M., Schreier, R., and Dietz, V., 2000, "Treadmill Training of Paraplegic Patients Using a Robotic Orthosis," J. Rehabil. Res. Dev., **37**(6), pp. 693–700.

[9] Veneman, J. F., Kruidhof, R., Hekman, E. E., Ekkelenkamp, R., Van Asseldonk, E. H., and Van Der Kooij, H., 2007, "Design and Evaluation of the LOPES Exoskeleton Robot for Interactive Gait Rehabilitation," IEEE Trans. Neural Syst. Rehabil. Eng., **15**(3), pp. 379–386.

[10] Banala, S. K., Kim, S. H., Agrawal, S. K., and Scholz, J. P., 2008, "Robot Assisted Gait Training With Active Leg Exoskeleton (ALEX)," Second IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics, Scottsdale, AZ, Oct. 19–22, pp. 653–658.

[11] Hesse, S., and Uhlenbrock, D., 2000, "A Mechanized Gait Trainer for Restoration of Gait," J. Rehabil. Res. Dev., **37**(6), pp. 701–708.

[12] Schmidt, H., Werner, C., Bernhardt, R., Hesse, S., and Krüger, J., 2007, "Gait Rehabilitation Machines Based on Programmable Footplates," J. Neuroeng. Rehabil., **4**(1), p. 2.

[13] Hesse, S., Waldner, A., and Tomelleri, C., 2010, "Innovative Gait Robot for the Repetitive Practice of Floor Walking and Stair Climbing Up and Down in Stroke Patients," J. Neuroeng. Rehabil., **7**(1), p. 30.
[14] Díaz, I., Gil, J. J., and Sánchez, E., 2011, "Lower-Limb Robotic Rehabilitation: Literature Review and Challenges," J. Rob., **2011**, p. 759764.
[15] Shao, Y., Xiang, Z., Liu, H., and Li, L., 2016, "Conceptual Design and Dimensional Synthesis of Cam-Linkage Mechanisms for Gait Rehabilitation," Mech. Mach. Theory, **104**, pp. 31–42.
[16] Tsuge, B. Y., and McCarthy, J. M., 2016, "An Adjustable Single Degree-of-Freedom System to Guide Natural Walking Movement for Rehabilitation," ASME J. Med. Devices, **10**(4), p. 044501.
[17] Deshpande, S., and Purwar, A., 2017, "A Task-Driven Approach to Optimal Synthesis of Planar Four-Bar Linkages for Extended Burmester Problem," ASME J. Mech. Rob., **9**(6), p. 061005.
[18] Ge, Q. J., Purwar, A., Zhao, P., and Deshpande, S., 2016, "A Task Driven Approach to Unified Synthesis of Planar Four-Bar Linkages Using Algebraic Fitting of a Pencil of G-Manifolds," ASME J. Comput. Inf. Sci. Eng., **17**(3), p. 031011.
[19] Hodgins, D., 2008, "The Importance of Measuring Human Gait," Med. Device Technol., **19**(5), pp. 42–44.
[20] Tahir, N. M., and Manap, H. H., 2012, "Parkinson Disease Gait Classification Based on Machine Learning Approach," J. Appl. Sci., **12**(2), pp. 180–185.
[21] Yang, M., Zheng, H., Wang, H., and McClean, S., 2009, "Feature Selection and Construction for the Discrimination of Neurodegenerative Diseases Based on Gait Analysis," Third International Conference on Pervasive Computing Technologies for Healthcare, London, UK, Apr. 1–3, pp. 1–7.
[22] Serrao, M., Chini, G., Bergantino, M., Sarnari, D., Casali, C., Conte, C., Ranavolo, A., Marcotulli, C., Rinaldi, M., Coppola, G., Bini, F., Pierelli, F., and Marinozzi, F., 2018, "Identification of Specific Gait Patterns in Patients With Cerebellar Ataxia, Spastic Paraplegia, and Parkinson's Disease: A Non-Hierarchical Cluster Analysis," Human Mov. Sci., **57**, pp. 267–279.
[23] Serrao, M., Chini, G., Bergantino, M., Sarnari, D., Casali, C., Conte, C., Ranavolo, A., Marcotulli, C., Rinaldi, M., Coppola, G., Bini, F., Pierelli, F., and Marinozzi, F., 2018, "Dataset on Gait Patterns in Degenerative Neurological Diseases," Data Brief, **16**, pp. 806–816.
[24] Ren, L., Jones, R. K., and Howard, D., 2007, "Predictive Modelling of Human Walking Over a Complete Gait Cycle," J. Biomech., **40**(7), pp. 1567–1574.
[25] Mannini, A., and Sabatini, A. M., 2014, "Walking Speed Estimation Using Foot-Mounted Inertial Sensors: Comparing Machine Learning and Strap-Down Integration Methods," Med. Eng. Phys., **36**(10), pp. 1312–1321.
[26] Williamson, R., and Andrews, B. J., 2000, "Gait Event Detection for FES Using Accelerometers and Supervised Machine Learning," IEEE Trans. Rehabil. Eng., **8**(3), pp. 312–319.
[27] Aertbeliën, E., and De Schutter, J., 2014, "Learning a Predictive Model of Human Gait for the Control of a Lower-Limb Exoskeleton," Fifth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, Sao Paulo, Brazil, Aug. 12–15, pp. 520–525.
[28] Luu, T. P., Lim, H. B., Hoon, K. H., Qu, X., and Low, K., 2011, "Subject-Specific Gait Parameters Prediction for Robotic Gait Rehabilitation Via Generalized Regression Neural Network," IEEE International Conference on Robotics and Biomimetics, Karon Beach, Phuket, Thailand, Dec. 7–11, pp. 914–919.
[29] Yun, Y., Kim, H.-C., Shin, S. Y., Lee, J., Deshpande, A. D., and Kim, C., 2014, "Statistical Method for Prediction of Gait Kinematics With Gaussian Process Regression," J. Biomech., **47**(1), pp. 186–192.
[30] Jolliffe, I., 2011, *Principal Component Analysis*, Springer-Verlag, New York.
[31] Fushiki, T., 2011, "Estimation of Prediction Error by Using K-Fold Cross-Validation," Stat. Comput., **21**(2), pp. 137–146.
[32] Han, S., Qubo, C., and Meng, H., 2012, "Parameter Selection in SVM With RBF Kernel Function," World Automation Congress, Puerto Vallarta, Mexico, June 24–28, pp. 1–4.
[33] Amari, S.-I., and Wu, S., 1999, "Improving Support Vector Machine Classifiers by Modifying Kernel Functions," Neural Networks, **12**(6), pp. 783–789.
[34] Haykin, S., 1994, *Neural Networks*, **2**, Prentice Hall, New York.
[35] Kingma, D. P., and Ba, J., 2014, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980.
[36] Li, Y., and Yuan, Y., 2017, "Convergence Analysis of Two-Layer Neural Networks With ReLU Activation," Adv. Neural Inf. Process. Syst., pp. 597–607.
[37] Van der Aalst, W. M., Rubin, V., Verbeek, H., van Dongen, B. F., Kindler, E., and Günther, C. W., 2010, "Process Mining: A Two-Step Approach to Balance Between Underfitting and Overfitting," Software Syst. Model., **9**(1), p. 87.
[38] Kingma, D. P., and Welling, M., 2014, "Auto-Encoding Variational Bayes," Proceedings of the 2nd International Conference on Learning Representations (ICLR), Banff, AB, Canada, Apr. 14–16.
[39] Kullback, S., and Leibler, R. A., 1951, "On Information and Sufficiency," Ann. Math. Stat., **22**(1), pp. 79–86.
[40] Radford, A., Metz, L., and Chintala, S., 2016, "Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks," Comp. Res. Reposit.