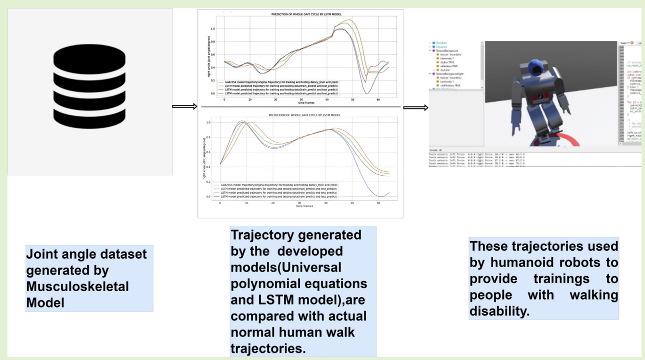
Development of the LSTM Model and Universal Polynomial Equation for All the Sub-Phases of Human Gait

Vijay Bhaskar Semwal, *Senior Member, IEEE*, Youngshik Kim, Vishwanath Bijalwan, *Senior Member, IEEE*, Astha Verma, Ghanapriya Singh, Neha Gaud, Hangyeol Baek, *Graduate Student Member, IEEE*, and Abdul Manan Khan, *Senior Member, IEEE*

***Abstract*—Human gait can be measured using a change in the joint angles value of the lower limb joints during walking. Inverse kinematics (IK) of a human leg refers to calculating joint angle values from different leg positions during locomotion. This article calculates a real-time IK for a three-link kinematic leg using a musculoskeletal model in OpenSim. The model provides the analytical solution of IK, which is quick and accurate. But, the model-based approach is very much dependent on the morphology of the bipedal robot. The existing bipedal robots are mainly planned for walking on flat terrain and unconstrained environment. So, these limitations lead to the developing of a generalized polynomial equation for generating walking trajectories. The model-based approaches rely on the degree of similarity between the virtual human model and the actual human body, which does not fully describe the human body. Due to the difference in real and virtual models, the model-based approaches suffer during the optimization process of the boundary conditions and the mass-inertial parameters, leading to an unstable numerical solution. The learning-based method using long short-term memory (LSTM) model for gait generation is proposed to overcome the limitation of model-based approaches.**



***Index Terms*—Curve fitting, deep learning, gait analysis, humanoid robot, inverse kinematics (IK), Opensim, rehabilitation.**

I. Introduction

TO UNDERSTAND the problem of patients with a walking disability and provide them with rehabilitation training, gait training robots have been developed. These robots assist the patient in performing the gait cycle. Robot-based assistance is better than traditional manual artificial lower limb rehabilitation and medical equipment for rehabilitation, which does not meet the requirements of disabled people [1]. For these robots, motion planning is needed to train the model. Motion planning is done by generating trajectories through the human walking model. So, in a real-time system [2], the three-link manipulator can simulate the motion of normal human walking. The bipedal robot walk is simulated using the human walking model. In our study, Opensim [3] is used as a real-time system with a human walking model. Here, trajectories are generated for a regular human walk by solving inverse kinematics (IK). This method provides accurate joint angle results in a shorter amount of time [4]. Previously, for gait retraining, researchers used real-time direct kinematics (DK) to calculate joint angles assuming markers were attached to segments. Joint angles are computed as biofeedback variables instead of IK. IK globally optimizes the generalized coordinates of the model to minimize marker tracking errors. The joint angles

Manuscript received 3 April 2023; accepted 22 May 2023. Date of publication 5 June 2023; date of current version 14 July 2023. This work was supported in part by the National Research Foundation of Korea (NRF) through the Korean Government (MSIT) under Grant 2022R1A2C1011462; and in part by the Ministry of Education, Government of India, through HEFA CSR, under Grant SAN/CSR/08/2021–22. The associate editor coordinating the review of this article and approving it for publication was Prof. Dongsoo Har. *(Corresponding authors: Vishwanath Bijalwan; Youngshik Kim.)*

Vijay Bhaskar Semwal is with the Department of CSE, MANIT Bhopal, Bhopal, 462003 India (e-mail: vsemwal@manit.ac.in).

Youngshik Kim, Vishwanath Bijalwan, Hangyeol Baek, and Abdul Manan Khan are with the Department of Mechanical Engineering, Hanbat National University, Daejeon 34158, South Korea (e-mail: youngshik@hanbat.ac.kr; vishwanath.bijalwan@itgopeshwar.ac.in).

Astha Verma is with Harbinger Systems Pvt. Ltd., Pune 411045, India (e-mail: astha2k12@gmail.com).

Ghanapriya Singh is with the Department of ECE, NIT Kurukshetra, Thanesar 136119, India.

Neha Gaud is with the Department of CS, DAVV Indore, Indore 452001, India.

Digital Object Identifier 10.1109/JSEN.2023.3281401

computed by DK are less accurate, and it takes more time for calculation [5].

This research work aims to generate human joint trajectories for a regular walk, which humanoid robots will use to provide rehabilitation training to people with a walking disability. The model-based method generates joint trajectories by solving IK in real-time. This method provides accurate joint angle values in a shorter amount of time. Even though many researchers used numerical methods like genetic algorithms, simulated annealing, etc., to solve IK earlier, these methods are less accurate and take more time to calculate.

This model-based method is for only those robots that can move on even ground surfaces. To generate trajectories for those robots, which can walk on any walking surface, we have also developed polynomial equations for all discrete sub-phases of human gait. So model-free walking trajectories can be generated using the polynomial equation, which is universal and generalized. Learning-based methods for human gait generation based on long short-term memory (LSTM) are presented in this work. The LSTM model is used to map gait parameters to gait patterns. The model will be trained using the human gait data generated in Opensim. So after training, it can generate reference gait patterns for a regular human walk.

A. Author's Contribution

1) The model-based method for solving IK in real-time is presented. This method provides accurate joint angle results in a shorter amount of time.
2) Design of the universal polynomial equations for all the sub-phases of human gait by using time-series gait data, which can act as alternatives for the human walk.
3) Validation of generated trajectories using polynomial equations over a humanoid robot, which performs a regular stable walk with these trajectories.
4) Development of a learning-based method for human gait generation based on an LSTM. The LSTM model generates reference gait patterns for motion planning for gait rehabilitation robots.

B. Organization of Article

This study is presented as follows. In Section II, the contributions of earlier researchers in solving IK are presented. Section III outlines the procedure for generating trajectories using various models. Section IV presents joint trajectories generated by both musculoskeletal models using polynomial equations and an LSTM model. Finally, Section V summarizes the work presented in this article.

II. Background and Significance

Researcher Nearchou et al. [6] presented a method for addressing the IK problem in redundant robots operating within complex environments by employing a modified genetic algorithm. This algorithm minimizes the end-effector's positional error and the robot's total joint displacement, ensuring safe free-space movement. Semwal et al. [7] proposed a solution for the IK problem using the Forward And Backward Reaching IK (FABRIK) method. Their work outlines the standard analytical solution for forward kinematics through the FK equation. They employ a displacement-based error minimization objective function in their approach, implementing a function to determine the intermediate angles for transitional points within a trajectory, spanning from the initial to the final position. In a subsequent investigation, Semwal et al. [8] presented the concept of utilizing a neural network (NN) to solve the IK problem specifically for a three-link robotic manipulator. Semwal et al. [9] introduced a method that combined automated systems and joint trajectory generation for bipedal walking. This approach utilized hybrid automata for creating trajectories during bipedal motion. State-dependent polynomial functions were generated for each stage of human gait [34], defining trajectories for six leg joints during bipedal movement [35]. Park et al. [10] suggested a technique to address the leg's IK by adapting the modified Improved Jacobian Pseudoinverse (mIJP). Hong [11] developed an approach based on Gaussian processes for learning trajectories and creating customized gait motions at variable walking speeds determined by the user. Yadav et al. [12] presented a method for crafting available bipedal robot walking trajectories using sinusoidal and cubic spline functions. Wong et al. [13] presented the development of a dynamic model, dubbed the three-mass linear inverted pendulum plus flywheel model (TLIPFM), to simulate the walking motion of humanoid robots. Also, Wu and Li [14] introduced the idea of a fuzzy dynamic gait pattern generator that allows humanoid robots to create appropriate gait patterns in real-time. An accelerometer and pressure sensors are combined through a fuzzy controller to rapidly react to external forces and produce an appropriate gait pattern. Alharbi et al. [15] proposed designing a NN to forecast future gait angles at fixed speeds using the Levenberg–Marquardt method. A gait data prediction model can be applied to create prosthetics and orthotics for individuals with lower limb issues [32], [33]. A NN has been developed to predict future gait angles at fixed speeds. Li et al. [16] presented a linear pendulum model (LPM) for trajectory planning. Researchers have employed model-based methods for trajectory planning; however, these approaches have limitations. Specifically, they are only applicable to existing humanoid robots that can move on uneven ground surfaces. Consequently, it is challenging to develop accurate human-like models for such robots [30], [31], [36]. To address this limitation, we proposed a universal polynomial equation for generating trajectories that any robot can employ. These polynomial equations are generated using the joint angle dataset obtained by solving IK with the Gait 2354 musculoskeletal model. Model-based methods have additional limitations, such as their output being heavily reliant on the degree of similarity between the natural human body and the virtual human body model. Furthermore, these methods cannot completely represent the natural human body. Another challenge is that mass inertial parameters and boundary conditions necessitating numerical stability influence the optimization process.

III. Experimental Design

Initially, real-world human walking data in Cartesian coordinates are supplied to a musculoskeletal model capable of simulating human locomotion. Thereafter, a dataset of joint angles is produced by solving the IK problem using the OpenSim software and the musculoskeletal model. Following this, we create a universal polynomial equation for generating motion trajectories. Upon comparing the trajectories generated by the polynomial equations to the actual human-like trajectories produced by two distinct musculoskeletal models (Gait 2354 and Gait 2392), it is observed that the polynomial equation-derived trajectories closely resemble the actual human-like trajectories. This confirms the precision of the developed polynomial equations. Providing the joint angle dataset generated by the polynomial equations to a humanoid robot allows it to achieve stable walking by adhering to these trajectories [17]. This further validates the correctness of our equations.

In order to address the issue with the model-based approach mentioned in the literature review, a learning-based technique employing trajectory generation is utilized. We develop an LSTM NN for this purpose. The training and testing of this model is carried out using a joint angle dataset generated in OpenSim by solving the IK problem with the Gait 2354 musculoskeletal model. The predicted trajectories from the LSTM model are then compared to the actual Gait 2354 musculoskeletal model trajectories. The similarities observed between the two sets of trajectories validate the accuracy of our LSTM model in predicting normal human walking patterns.

A. Method for Solving IK of Musculoskeletal Model in OpenSim

In Opensim, for every time frame, we calculate IK using the human walking data, which is experimentally collected. It computes joint angles for a walking model by reducing the weighted sum of marker errors. OpenSim solves a weighted least squares optimization problem to minimize marker error. Marker error is the distance between the location of the original (experimental) marker and the corresponding model marker location. Each marker has its weight value, which indicates how much that marker's error term should be reduced in the least squares problem. This equation can express the minimization of the weighted sum of marker errors

$$\min_q \left[\sum_{i \in \text{ markers}} w_i \left\| x_i^{\exp} - x_i(q) \right\|^2 \right] \tag{1}$$

where q = vector of calculated joint angles, $x_i^{\exp}$ = cartesian position of the experimental marker i.

$x_i(q)$ = cartesian position of the corresponding model marker i (it depends on q), w_i = weight value of marker i.

1) Input Dataset: The experimental marker file is the input dataset containing markers placed on the model's body segment. These markers describe the cartesian position of different leg segments during [18] motion.

2) Output Dataset: It shows the motion file generated after solving IK. This file contains six leg joint angles (left and right ankle, hip, and knee) varying with time during motion. This data is for one gait cycle ranging from 0.4 to 1.6 s at 1 m/s speed. Algorithm 1 explains the procedure to solve IK [38] for gait analysis which is a standard procedure available in opensim.

Algorithm 1 Solving IK

Input: Using gait2354_simbody.osim file - the standard musculoskeletal model gait 2354
Using gait2354_Setup_IK.xml file- which is pre-configured settings in opensim for solving IK.
Using subject01_walk1.trc file - Experimental marker trajectory file.
Output: Using subject01_walk1_ik.mot - IK solution will be saved in motion file.
Begin:
1) Load model - loading musculoskeletal model (gait2354_simbody.osim).
2) Scale model - Provide Scale setup xml file (gait2354_Setup_Scale.xml) to scale the model. The scaled model have name (subject01_simbody.osim) with model markers.
3) Setup Inverse Kinematics analysis - Provide IK tool setup file (subject01_Setup_IK.xml) to the model, which loads both files: (subject01_walk1.trc), (subject01_walk1_ik.mot)
4) Solve Inverse Kinematics problem - during the movement of the model for one whole stride, the IK problem is start to solve.

End

Algorithm 2 explains the process of generating polynomial equations.

Algorithm 2 Creation of Polynomial Equations

Input: Time series gait data (Gait2354) comprises of all the joint angles of human leg, which i changing over time.
Output: Polynomial functions corresponding to each sub-phase of the stride.
Begin:
1) The Gait2354 model's time series gait data is separated into gait phases.
We proceed as follows for each data phase:
   a) To fit this, we use the **function Polyfit**
   Polyfit(x, y, n).
      i) Input to function
      x and y - the x and y axis values of the data points that we wish to fit.
      The degree of our polynomial function is specified by n.
      ii) Output of the function
      By minimizing the sum of the squares of the deviations of the provided y data from the model, this function finds the coefficients of a polynomial p(x) of degree n that matches the given y data.
   b) **Poly 1d function**-
   Create a polynomial function using the estimated coefficients.
   c) **Polyval function**
   Polyval(coefficients, x)
   To construct a curve to fit data, utilize the coefficients produced by the above polyfit function to estimate a new set of y values for a given set of x values.
2) During one complete stride, a new joint angle dataset and trajectories produced by the polynomial equation -
   a) Eventually, for each joint angle, these polynomial equations created new joint angle (ynew) values for distinct phases, which were then merged to form data for a single gait cycle.
   b) Then, for one whole stride, trajectories are generated.
   c) Finally, for one entire stride, these trajectories are compared to the original data trajectories.

EndBegin

B. Polynomial Modeling of Gait Cycle Phases

The time series gait data of the Gait2354 model is divided into different phases of gait [19]. The divided data is then used for generating equations in Python [20]. The equations are generated using curve fitting [21]. Here equations are generated for each discrete phase of the gait cycle [1-loading response (LR), 2-mid stance (MS), 3-terminal stance (TS), 4-pre-swing (PS), 5-initial swing (IS), 6-mid swing (MSw), 7-terminal swing (TSw)] [37], so to get less error in each phase and more accuracy, polynomial equations of degrees

TABLE I
POLYNOMIAL EQUATION OF LEFT ANKLE JOINT

Function:	$f(x)=a_1x^8+a_2x^7+a_3x^6+a_4x^5+a_5x^4+a_6x^3+a_7x^2+a_8x+a_9$							
LR:	$a_1=5.85$	$a_2=-1.32$	$a_3=1.19$	$a_4=-5.39$	$a_5=1.21$	$a_6=-1.08$		
PS:	$a_1=-2.01$	$a_2=9.55$	$a_3=-1.80$	$a_4=1.71$	$a_5=-8.07$	$a_6=1.52$		
MS:	$a_1=3.10$	$a_2=-1.49$	$a_3=3.13$	$a_4=-3.76$	$a_5=2.81$	$a_6=-1.34$	$a_7=4.00$	$a_8=-6.79$	$a_9=5.03$
TS:	$a_1=4.40$	$a_2=-2.37$	$a_3=5.46$	$a_4=-6.94$	$a_5=5.28$	$a_6=-2.39$	$a_7=6.01$	$a_8=-6.44$	
MSw:	$a_1=2.16$	$a_2=-1.64$	$a_3=5.09$	$a_4=-8.19$	$a_5=6.90$	$a_6=-2.43$	$a_7=2.29$	$a_8=2.81$	
IS:	$a_1=8.77$	$a_2=-7.25$	$a_3=2.38$	$a_4=-4.04$	$a_5=3.76$	$a_6=-1.83$	$a_7=3.65$		
TSw:	$a_1=-5.11$	$a_2=4.54$	$a_3=-1.68$	$a_4=3.3$	$a_5=-3.67$	$a_6=2.17$	$a_7=-5.34$		

TABLE II
POLYNOMIAL EQUATION OF RIGHT ANKLE JOINT

Function:	$f(x)=a_1x^6+a_2x^5+a_3x^4+a_4x^3+a_2x+a_6x+a_7$					
LR	$a_1=4.17$	$a_2=-7.37$	$a_3=4.85$	$a_4=-1.4$	$a_5=1534$		
LR:	$a_1=-1.46$	$a_2=4.34$	$a_3=-5.15$	$a_4=3.04$	$a_5=-8.93$	$a_6=1.04$	
TS	$a_1=-4.39$	$a_2=2.08$	$a_3=-3.96$	$a_4=3.76$	$a_5=-1.78$	$a_6=3.38$	
PS:	$a_1=3.2$	$a_2=-1.51$	$a_3=2.87$	$a_4=-2.72$	$a_5=1.28$	$a_6=-2.43$	
MSw	$a_1=2.16$	$a_2=-1.40$	$a_3=3.64$	$a_4=-4.70$	$a_5=3.03$	$a_6=-7.82$	
IS	$a_1=-2386$	$a_2=7698$	$a_3=-5002$	$a_4=-7524$	$a_5=1.12$	$a_6=-3999$	
MS:	$a_1=-1.63$	$a_2=6.09$	$a_3=-9.4$	$a_4=7.66$	$a_5=-3.4$	$a_6=8.40$	$a_7=-8.37$
TSw:	$a_1=-5.32$	$a_2=4.80$	$a_3=-1.80$	$a_4=3.62$	$a_5=-4.07$	$a_6=2.44$	$a_7=-6.10$

TABLE III
POLYNOMIAL EQUATION OF LEFT HIP JOINT

Function	$f(x)=a_1x^8+a_2x^7+a_3x^6+a_4x^5+a_5x+a_4+a_6x^3+a_7x^2+a_8x+a_9$								
LR:	$a_1=2.69$	$a_2=-7.24$	$a_3=8.11$	$a_4=-4.83$	$a_5=1.62$	$a_6=-2.89$	$a_7=2.14$		
MS:	$a_1=4.54$	$a_2=2.17$	$a_3=4.55$	$a_4=-5.42$	$a_5=4.03$	$a_6=-1.91$	$a_7=5.66$	$a_8=-9.55$	$a_9=7.03$
IS:	$a_1=1.75$	$a_2=-1.62$	$a_3=6.59$	$a_4=-1.52$	$a_5=2.19$	$a_6=-2.02$	$a_7=1.16$	$a_8=-3.80$	$a_9=5.44$
MSw:	$a_1=4.03$	$a_2=-4.19$	$a_3=1.90$	$a_4=-4.94$	$a_5=8.03$	$a_6=-8.33$	$a_7=5.39$	$a_8=-1.99$	$a_9=3.23$
TSw:	$a_1=-3.57$	$a_2=4.16$	$a_3=-2.11$	$a_4=6.15$	$a_5=-1.11$	$a_6=1.28$	$a_7=-9.28$	$a_8=3.81$	$a_9=-6.84$
TS:	$a_1=5497$	$a_2=9497$	$a_3=-6.96$	$a_4=1.02$	$a_5=-6.02$	$a_6=1.28$			
PS:	$a_1=5.12$	$a_2=-2.42$	$a_3=4.60$	$a_4=-4.36$	$a_5=2.06$	$a_6=-3.91$			

TABLE IV
POLYNOMIAL EQUATION OF RIGHT HIP JOINT

Function:	$f(x)=a_1x^8+a_2x^7+a_3x^6+a_4x^5+a_5x^4+a_6x^3+a_7x^2+a_8x+a9$								
LR:	$a_1=3.64$	$a_2=-9.83$	$a_3=1.10$	$a_4=-6.59$	$a_5=2.21$	$a_6=-3.96$	$a_7=2.94$	$a_8=0$	$a_9=0$
MSw:	$a_1=-3.8$	$a_2=2.95$	$a_3=-9.58$	$a_4=1.65$	$a_5=-1.60$	$a_6=8.31$	$a_7=-1.79$	$a_8=0$	$a_9=0$
MS:	$a_1=-3.49$	$a_2=1.67$	$a_3=-3.51$	$a_4=4.19$	$a_5=-3.12$	$a_6=1.49$	$a_7=-4.42$	$a_8=7.50$	$a_9=-5.55$
TS:	$a_1=-2.93$	$a_2=1.24$	$a_3=-2.09$	$a_4=1.74$	$a_5=-7.23$	$a_6=1.19$	$a_7=0$	$a_8=0$	$a_9=0$
TSw:	$a_1=-8.84$	$a_2=6.57$	$a_3=-1.95$	$a_4=2.9$	$a_5=-2.15$	$a_6=6.38$	$a_7=0$	$a_8=0$	$a_9=0$
PSw:	$a_1=1.30$	$a_2=-6.19$	$a_3=1.17$	$a_4=-1.11$	$a_5=5.27$	$a_6=-9.99$	$a_7=0$	$a_8=0$	$a_9=0$
IS:	$a_1=1.15$	$a_2=-6.35$	$a_3=1.39$	$a_4=-1.52$	$a_5=8.33$	$a_6=-1.82$	$a_7=0$	$a_8=0$	$a_9=0$

that vary from second order to eighth order are generated [23]. The polynomial functions for all the joints and all the seven phases of the human gait cycle are given below. Tables I–VI shows the generated polynomial equation and coefficients of the left ankle joint, right ankle joint, left hip joint, right hip joint, left knee joint, right knee joint for different sub-phase respectively.

C. Designing an LSTM-Based Gait Generation System

RNN is used for dynamic information processing like time series prediction, processing control, etc. [26] RNNs are NNs designed for modeling time series data like gait patterns [27]. RNNs are naturally suitable for gait generation tasks. In RNN, for output prediction at the current time "t," both information (new current information from the input layer and information from the predicted output at the previous time step) are required as input to the hidden layer and this process is repeated. So, our RNN model uses a LSTM, a variant of the traditional RNN [28], [29], as our gait generator, which takes gait parameters as input and outputs gait patterns. Here, the LSTM was trained on 47 samples from the normal human walk dataset and tested on 26 samples of the 23 kinematic input variables (i.e., joint angles). The LSTM model learned the lower limb kinematic trajectories using the training samples and tested them with other samples from our normal human walk dataset. Algorithm 3 explains the procedure to

TABLE V
POLYNOMIAL EQUATION OF LEFT KNEE JOINT

Function:	$f(x)=a_1x^5+a_2x^4+a_3x^3+a_4x^2+a_5x+a_6$					
loading response:	$a_1=-4.27$	$a_2=9.66$	$a_3=-8.70$	$a_4=3.04$	$a_5=-8.79$	$a_6=7.88$
MS	$a_1=-1.46$	$a_2=4.34$	$a_3=-5.15$	$a_4=3.04$	$a_5=-8.93$	$a_6=1.04$
PS	$a_1=-4.39$	$a_2=2.08$	$a_3=-3.96$	$a_4=3.76$	$a_5=-1.78$	$a_6=3.38$
IS	$a_1=-4.26$	$a_2=2.19$	$a_3=-4.54$	$a_4=4.76$	$a_5=-2.52$	$a_6=5.37$
TS	$a_1=9.22$	$a_2=-4.16$	$a_3=7.46$	$a_4=-6.64$	$a_5=2.93$	$a_6=-5.11$
MSw	$a_1=1.44$	$a_2=-1.19$	$a_3=3.85$	$a_4=-6.04$	$a_5=4.63$	$a_6=-1.4$
TSw	$a_1=9.04$	$a_2=-6.75$	$a_3=2.01$	$a_4=-3.00$	$a_5=2.23$	$a_6=-6.65$

TABLE VI
POLYNOMIAL EQUATION OF RIGHT KNEE JOINT

Function:	$f(x)=a_1x^8+a_2x^7+a_3x^6+a_4x^5+a_5x^4+a_6x^3+a_7x^2+a_8x+a_9$							
LR:	$a_1=-2178$	$a_2=3620$	$a_3=-1576$	$a_4=135.3$				
MS:	$a_1=-4.17$	$a_2=1.30$	$a_3=-1.62$	$a_4=9.96$	$a_5=-3.02$	$a_6=3.61$		
PS:	$a_1=-1.12$	$a_2=5.32$	$a_3=-1.01$	$a_4=9.57$	$a_5=-4.53$	$a_6=8.59$		
TS:	$a_1=8.05$	$a_2=-3.25$	$a_3=5.24$	$a_4=-4.18$	$a_5=1.65$	$a_6=-2.59$		
MSw:	$a_1=2.86$	$a_2=-1.83$	$a_3=4.71$	$a_4=-6.03$	$a_5=3.86$	$a_6=-9.88$		
TSw:	$a_1=7.04$	$a_2=-5.32$	$a_3=1.61$	$a_4=-2.43$	$a_5=1.83$	$a_6=-5.15e+05$		
IS:	$a_1=1.4e$	$a_2=-1.23$	$a_3=4.75$	$a_4=-1.04$	$a_5=1.43$	$a_6=-1.26$	$a_7=6.94$	$a_8=-2.17$ $a_9=2.98$

predict gait patterns for a normal human walk by the LSTM model.

D. Tuning of Hyperparameters and Training of Model

So first, we tuned three hyperparameters (number of hidden layers, number of neurons in each hidden layer, and learning rate), which was done with the help of the Keras Tuner library. From the Keras tuner, we import the RandomSearch library [for regression, RandomSearch is used]. Random Search is a technique that uses random hyperparameter combinations to find the best solution for the built model. Here, we provide a range of a number of particular hyperparameters required, and random combinations of hyperparameters are considered in every iteration. Then, a random search will find optimal parameters with the lowest mean absolute error. So here, when we create our sequential model, We provide a number of hidden layers ranging from 2 to 20. We provide a number of neurons (inside each hidden layer) ranging from 32 to 512.

Our study conducted five trials, with three executions in each trial. Each execution consisted of five epochs, and we applied RandomSearch to minimize the validation mean absolute error in each trial. The validation loss dropped during this process with each epoch. The five best trials were chosen based on the set of hyperparameter settings that produced the lowest validation mean absolute error score. By examining the model summary and choosing the trial with the lowest validation mean absolute error, we determined the optimal trial (0.3296). The model was then trained using the xtrain and ytrain datasets, utilizing the hyperparameter values from this experiment. The xtest and ytest datasets were used to validate the model, and its performance was assessed by showing the training and validation loss curves. Together with tuning the other hyperparameters, we increased the value of the epochs-number hyperparameter from 100 to 500. The model's training and validation data displayed a declining trend that stabilized around 500 epochs. This shows that the model was well-trained. Fig. 1 shows the simulation of the HOAP robot. Figs. 2–4 shows Gait Pattern of left and right hip, knee and ankle joints respectively for Gait2354 musculoskeletal model (blue) and LSTM model predicted trajectories (orange, green, red). Here, after scaling of data using MinMaxScaler, our dataset will be having joint angle values (Y-axis) between 0 and 1.

Algorithm 3 Prediction of Trajectories by LSTM Model

Input: Time series gait data collected for normal human walk which is generated in opensim by solving IK.
Output: Prediction of trajectories by LSTM model.
Begin:
1) Data is scaled by using MinMaxScaler. After scaling, our dataset will be having joint angle values between 0 to 1.
2) Then dataset is splitted into train and test data. (47 samples for training and 26 samples for testing).
3) Sequential LSTM model is created by using Keras library from Tensorflow 2.4.1. Rectified linear activation function (RELU) is used.
4) Tuning of Hyperparameters is done with the help of keras tuner library. Combination of best hyperparameters values (which provide minimum validation mean absolute error value (score)) provided by random search are:
   a) number of hidden LSTM layers =4
   b) First hidden layer has 448 neurons,
      2nd hidden layer has 384 neurons,
      3rd hidden layer has 128 neurons,
      4th hidden layer has 224 neurons,
   c) learning rate=.001
5) This best model will be trained with Adam optimizer.
6) Then xtest data is provided to the model and it will do **time series prediction** for this dataset by looking back into 3-time steps into the history for each time frame. Then we will compare plotted trajectories of this predicted data with trajectories generated from original data (ytest). We observed that original and predicted trajectories are similar, so our created model for gait generation is correct.

EndBegin

IV. RESULTS AND DISCUSSION

Table VII shows the joint angle dataset for all six joints generated by polynomial equations. The trajectories of actual gait (musculoskeletal models Gait 2354 and Gait 2392) and polynomial function generated trajectories are compared. It is observed that joint trajectories determined by polynomial equations are the same as the actual normal human walk-

TABLE VII
JOINT ANGLE DATASET GENERATED BY POLYNOMIAL EQUATION

time	hip_flexion_r	knee_angle_r	ankle_angle_r	hip_flexion_l	knee_angle_l	ankle_angle_l
0.4	20.16323132	-55.19599817	2.738419	-5.79549239	-7.33364098	5.39685143
0.417	20.61644618	-50.34765447	2.74010352	-7.2599259	-6.33404953	5.62575207
0.433	20.96406359	-45.00070363	2.64647266	-8.75844852	-5.3167808	5.89146145
0.45	20.78903615	-39.21564442	2.37904368	-10.30761724	-4.47841451	6.24675532
0.467	20.78238376	-33.05297562	1.93658585	-11.46702135	-3.84790562	6.63812456
0.483	20.14990523	-26.57319602	1.39512056	-13.0008006	-2.9398077	6.99615157
0.5	19.64774636	-19.83680439	0.90792125	-14.38123778	-2.07650312	7.32588666
0.517	19.1909365	-13.81063982	0.6521642	-15.62228035	-1.46152018	7.76063997
0.533	18.62881768	-8.0705783	0.62084993	-16.87239837	-1.13023352	8.28113479

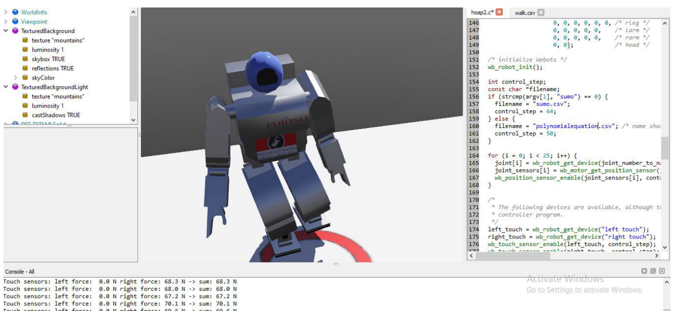
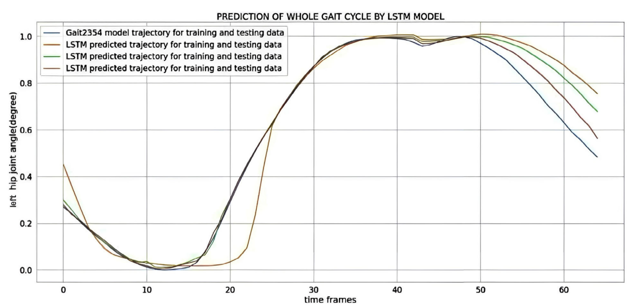
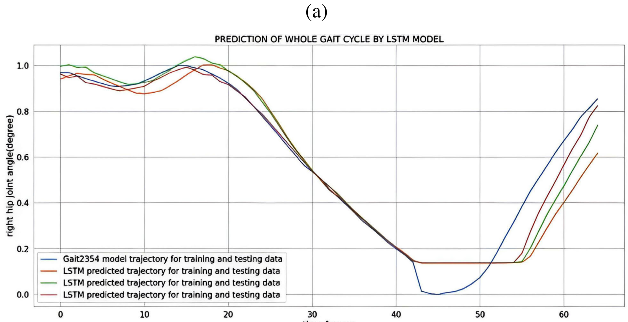


Fig. 1. Simulation of HOAP robot.



(a)



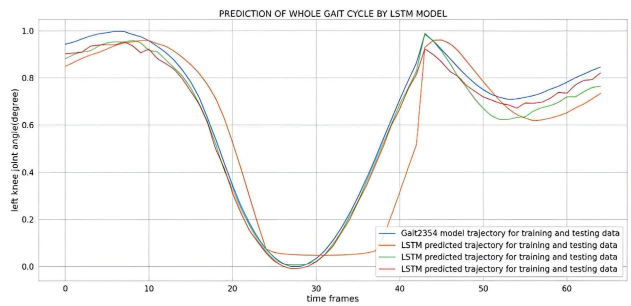
(b)

Fig. 2. Gait pattern of left and right hip using Gait2354 model and LSTM model for three different learning parameters 0.1, 0.01, and 0.001. (a) Left hip joint trajectory. (b) Right hip joint trajectory.

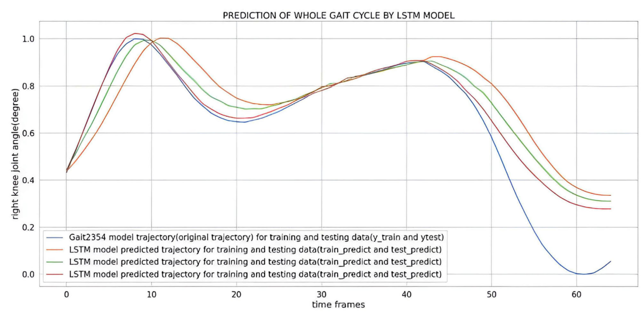
like trajectories of both musculoskeletal models [22]. So these are generalized equations with time as input [23]. So these equations are correct and can be used for finding joint trajectories for any robot [24].

A. Validation of Joint Trajectories Generated by Polynomial Equations With HOAP 2 Model

For four joint angles, left and right knee and hip joints, these generated values for different phases are combined to



(a)



(b)

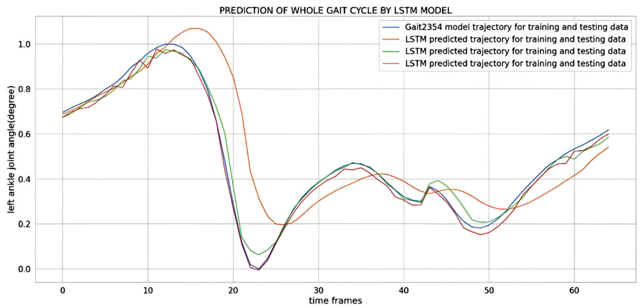
Fig. 3. Gait pattern of left and right knee using Gait2354 model and LSTM model for three different learning parameters 0.1, 0.01, and 0.001. (a) Left knee joint trajectory. (b) Right knee joint trajectory.

form data of one complete gait cycle and create a motion file (.csv). This motion file data of four joint angles are provided to this robot. It is observed that this humanoid [25] model walks in a stable manner using the trajectories generated from the developed polynomial equations. As a result, these trajectories are correct, demonstrating that the developed polynomial equations are correct.

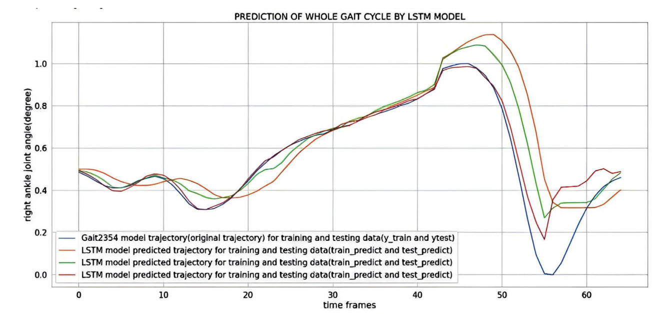
B. Trajectory of Actual Gait (Musculoskeletal Model Gait2354) and LSTM Model Predicted Trajectory Comparison

It is observed that we will get the same joint trajectories from the LSTM model as the actual human-like trajectories of the OpenSim musculoskeletal model. So the LSTM model is able to generate the correct reference gait pattern for a normal human walk.

1) Performance Analysis of Created LSTM Model: After training the LSTM model, the performance of our RNN model is evaluated using training and validation loss curves. We observed from the graph that training and validation loss

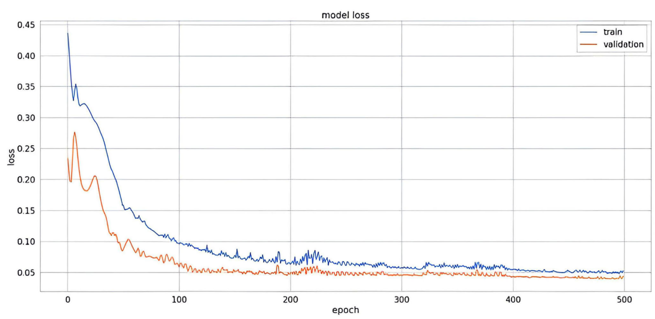


(a)

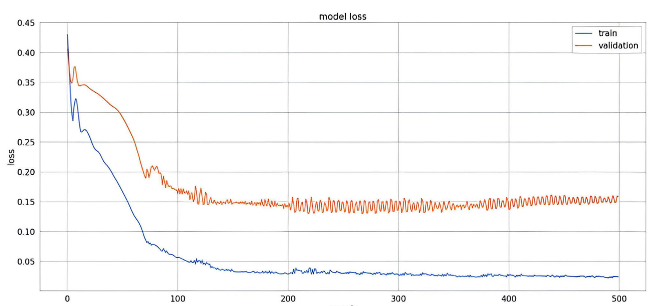


(b)

Fig. 4. Gait pattern of left and right ankle using Gait2354 model and LSTM model for three different learning parameters 0.1, 0.01, and 0.001. (a) Left ankle joint trajectory. (b) Right ankle joint trajectory.

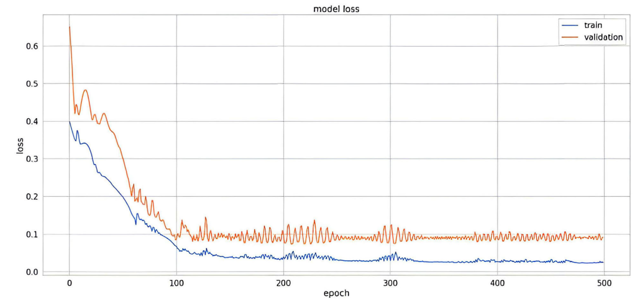


(a)

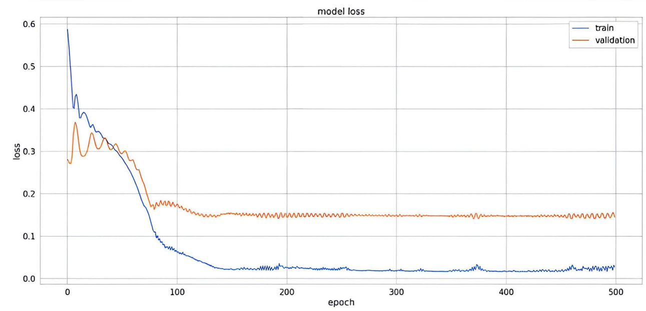


(b)

Fig. 5. Loss curve of left and right ankle using LSTM. (a) Left ankle joint loss curve. (b) Right ankle loss curve.

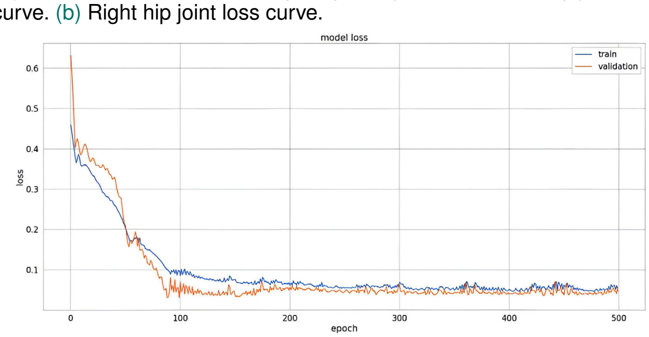


(a)

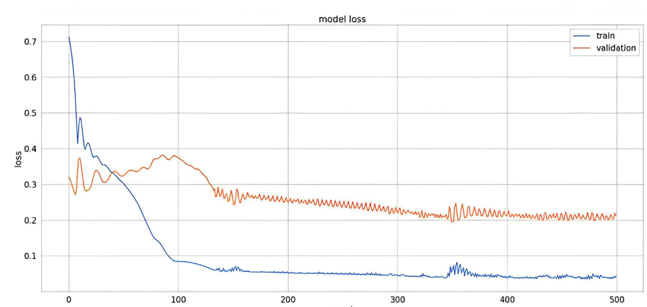


(b)

Fig. 6. Loss curve of left and right hip using LSTM. (a) Left hip joint loss curve. (b) Right hip joint loss curve.



(a)



(b)

Fig. 7. Training loss (blue) and validation loss (red) curve for left and right joints using LSTM model. (a) Left knee joint loss curve. (b) Right knee joint loss curve.

decreases and, at last, stabilizes around the same point. So, our LSTM model is well-trained. So the model's performance is good on both training and validation datasets. So our model is a good fit. Figs. 5–7 shows the training loss and validation loss curves for all the different joints during the model training. Here, the x-axis shows the number of epochs, and the y-axis shows a loss. During the training of the LSTM model, the training loss (blue) and validation loss (red) curves for the left and right ankle, hip, and knee were computed.

V. Conclusion and Future Research

A fast and accurate model-based IK solution for real-time calculation is presented to compute the joint angles. Different polynomial equations are generated for different sub-phases of gait. Furthermore, the walking trajectories are generated from developed polynomial equations. A comparison of these trajectories with both OpenSim model trajectories is presented. Validation on a real humanoid HOAP robot demonstrates that the equations we developed are correct. To provide a generalized technique to generate a gait pattern, a method based on recurrent NNs is developed. The gait data set that is generated in OpenSim is used to train an LSTM model, and then it predicts joint trajectories for a normal walk. We have achieved a very good performance.

By applying the inverse dynamics method, futuristic research directions may include computing net forces and torques at each joint to achieve a specific movement of joints. These joint trajectories, which we created by solving IK, are required as input to the model for manipulator dynamics. Because these stresses and torques regulate the movement of joints on biped robots, so the controller should be designed in a manner that resembles a regular human walk.

REFERENCES

[1] S. K. Challa, A. Kumar, V. B. Semwal, and N. Dua, "An optimized-LSTM and RGB-D sensor-based human gait trajectory generator for bipedal robot walking," *IEEE Sensors J.*, vol. 22, no. 24, pp. 24352–24363, Dec. 2022.

[2] P. Patil, K. S. Kumar, N. Gaud, and V. B. Semwal, "Clinical human gait classification: Extreme learning machine approach," in *Proc. 1st Int. Conf. Adv. Sci., Eng. Robot. Technol. (ICASERT)*, May 2019, pp. 1–6.

[3] J. A. Reinbolt, A. Seth, and S. L. Delp, "Simulation of human movement: Applications using OpenSim," *Proc. IUTAM*, vol. 2, pp. 186–198, Jan. 2011.

[4] V. B. Semwal, C. Kumar, P. K. Mishra, and G. C. Nandi, "Design of vector field for different subphases of gait and regeneration of gait pattern," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 104–110, Jan. 2018.

[5] C. Pizzolato, M. Reggiani, L. Modenese, and D. G. Lloyd, "Biofeedback for gait retraining based on real-time estimation of tibiofemoral joint contact forces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 9, pp. 1612–1621, Sep. 2017.

[6] A. C. Nearchou, "Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm," *Mechanism Mach. Theory*, vol. 33, no. 3, pp. 273–292, Apr. 1998.

[7] V. B. Semwal, M. Reddy, and A. Narad, "Comparative study of inverse kinematics using data driven and FABRIK approach," in *Proc. Adv. Robot.-5th Int. Conf. Robot. Soc.*, Jun. 2021, pp. 1–12.

[8] V. B. Semwal and Y. Gupta, "Performance analysis of data-driven techniques for solving inverse kinematics problems," in *Proc. SAI Intell. Syst. Conf.* Cham, Switzerland: Springer, 2022, pp. 85–99.

[9] V. B. Semwal and G. C. Nandi, "Generation of joint trajectories using hybrid automate-based model: A rocking block-based approach," *IEEE Sensors J.*, vol. 16, no. 14, pp. 5805–5816, Jul. 2016.

[10] H. Park, B. Kwak, and J. Bae, "Inverse kinematics analysis and COG trajectory planning algorithms for stable walking of a quadruped robot with redundant DOFs," *J. Bionic Eng.*, vol. 15, no. 4, pp. 610–622, Jul. 2018.

[11] J. Hong, C. Chun, S. Kim, and F. C. Park, "Gaussian process trajectory learning and synthesis of individualized gait motions," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 6, pp. 1236–1245, Jun. 2019, doi: 10.1109/TNSRE.2019.2914095.

[12] G. K. Yadav, S. Jaiswal, and G. C. Nandi, "Generic walking trajectory generation of biped using sinusoidal function and cubic spline," in *Proc. 7th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2020, pp. 745–750, doi: 10.1109/SPIN48934.2020.9071083.

[13] C. Wong, S. Xiao, and H. Aoyama, "Natural walking trajectory generator for humanoid robot based on three-mass LIPFM," *IEEE Access*, vol. 8, pp. 228151–228162, 2020.

[14] L.-F. Wu and T.-H.-S. Li, "Fuzzy dynamic gait pattern generation for real-time push recovery control of a teen-sized humanoid robot," *IEEE Access*, vol. 8, pp. 36441–36453, 2020.

[15] Y. Liu, S. Heng, X. Zang, Z. Lin, and J. Zhao, "Multiphase trajectory generation for planar biped robot using direct collocation method," *Math. Problems Eng.*, vol. 2021, pp. 1–14, Jan. 2021, doi: 10.1155/2021/6695528.

[16] L. Li, Z. Xie, X. Luo, and J. Li, "Trajectory planning of flexible walking for biped robots using linear inverted pendulum model and linear pendulum model," *Sensors*, vol. 21, no. 4, p. 1082, Feb. 2021, doi: 10.3390/s21041082.

[17] R. Jain, V. B. Semwal, and P. Kaushik, "Stride segmentation of inertial sensor data using statistical methods for different walking activities," *Robotica*, vol. 40, no. 8, pp. 2567–2580, Aug. 2022.

[18] S. Porsa, Y.-C. Lin, and M. G. Pandy, "Direct methods for predicting movement biomechanics based upon optimal control theory with implementation in OpenSim," *Ann. Biomed. Eng.*, vol. 44, no. 8, pp. 2542–2557, Aug. 2016.

[19] V. B. Semwal, R. Jain, P. Maheshwari, and S. Khatwani, "Gait reference trajectory generation at different walking speeds using LSTM and CNN," *Multimedia Tools Appl.*, vol. 2023, pp. 1–19, Mar. 2023.

[20] A. Gupta and V. B. Semwal, "Occluded gait reconstruction in multi person gait environment using different numerical methods," *Multimedia Tools Appl.*, vol. 81, no. 16, pp. 23421–23448, Jul. 2022.

[21] R. Jain and V. B. Semwal, "A novel feature extraction method for preimpact fall detection system using deep learning and wearable sensors," *IEEE Sensors J.*, vol. 22, no. 23, pp. 22943–22951, Oct. 2022.

[22] G. C. Nandi, V. B. Semwal, M. Raj, and A. Jindal, "Modeling bipedal locomotion trajectories using hybrid automata," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2016, pp. 1013–1018.

[23] V. B. Semwal, S. A. Katiyar, R. Chakraborty, and G. C. Nandi, "Biologically-inspired push recovery capable bipedal locomotion modeling through hybrid automata," *Robot. Auto. Syst.*, vol. 70, pp. 181–190, Aug. 2015.

[24] V. B. Semwal and G. C. Nandi, "Toward developing a computational model for bipedal push recovery—A brief," *IEEE Sensors J.*, vol. 15, no. 4, pp. 2021–2022, Apr. 2015.

[25] V. B. Semwal and G. C. Nandi, "Study of humanoid push recovery based on experiments," 2014, *arXiv:1405.4450*.

[26] S. A. Zargar, "Introduction to sequence learning models: RNN, LSTM, GRU," Dept. Mech. Aerosp. Eng., North Carolina State Univ., Raleigh, NC, USA, Tech. Rep., 2021. [Online]. Available: https://www.researchgate.net/publication/350950396_Introduction_to_Sequence_Learning_Models_RNN_LSTM_GRU?channel=doi&linkId=607b41c0907dcf667ba83ade&showFulltext=true, doi: 10.13140/RG.2.2.36370.99522.

[27] R. Jain, V. B. Semwal, and P. Kaushik, "Deep ensemble learning approach for lower extremity activities recognition using wearable sensors," *Expert Syst.*, vol. 39, no. 6, Jul. 2022, Art. no. e12743.

[28] A. Gupta and V. B. Semwal, "Multiple task human gait analysis and identification: Ensemble learning approach," in *Emotion and Information Processing*. Cham, Switzerland: Springer, 2020, pp. 185–197.

[29] N. S. Gaud and G. C. Nandi, "Human gait state prediction using cellular automata and classification using ELM," in *Machine Intelligence and Signal Analysis*. Singapore: Springer, 2019, pp. 135–145.

[30] A. Seth, M. Sherman, J. A. Reinbolt, and S. L. Delp, "OpenSim: A musculoskeletal modeling and simulation framework for in silico investigations and exchange," *Proc. IUTAM*, vol. 2, pp. 212–232, Jan. 2011.

[31] V. Bijalwan, V. B. Semwal, and T. K. Mandal, "Fusion of multi-sensor-based biomechanical gait analysis using vision and wearable sensor," *IEEE Sensors J.*, vol. 21, no. 13, pp. 14213–14220, Jul. 2021.

[32] V. B. Semwal, N. Gaud, P. Lalwani, V. Bijalwan, and A. K. Alok, "Pattern identification of different human joints for different human walking styles using inertial measurement unit (IMU) sensor," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 1149–1169, Feb. 2022.

[33] S. V. Bhaskar et al., "Speed, cloth and pose invariant gait recognition-based person identification," in *Machine Learning: Theoretical Foundations and Practical Applications*. Singapore: Springer, 2021, pp. 39–56.

[34] M. Raj, V. B. Semwal, and G. C. Nandi, "Hybrid model for passive locomotion control of a biped humanoid: The artificial neural network approach," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 5, no. 1, pp. 40–46, 2018.

[35] A. Verma, "Development of universal polynomial equation for all the sub-phases of human gait," in *Machine Vision and Augmented Intelligence Theory and Applications*. Singapore: Springer, 2021, pp. 45–55.

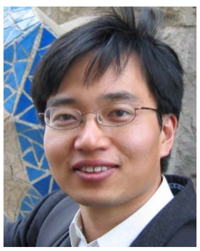
[36] E. M. Arnold, S. R. Ward, R. L. Lieber, and S. L. Delp, "A model of the lower limb for analysis of human movement," *Ann. Biomed. Eng.*, vol. 38, no. 2, pp. 269–279, Feb. 2010.

[37] V. Bhaskar Semwal, "Data driven computational model for bipedal walking and push recovery," 2017, *arXiv:1710.06548*.

[38] J. Hicks. *OpenSim Documentations: How Inverse Kinematics Works*. Accessed: Dec. 2020. [Online]. Available: https://simtk-confluence.stanford.edu

Vijay Bhaskar Semwal (Senior Member, IEEE) received the B.Tech. degree from the College of Engineering Roorkee, Roorkee, India, in 2008, and the M.Tech. and Ph.D. degrees from the Indian Institute of Information Technology Allahabad, Allahabad, India, in 2010 and 2017, respectively.

He was a Senior System Engineer (Research and Development) with Siemens, Gurugram, Haryana, and Bengaluru, India. He has also served as an Assistant Professor with the Department of CSE, NIT Rourkela, Rourkela, Odisha, India; IIIT Dharwad, Dharwad, India; and NIT Jamshedpur, Jamshedpur, India. He is currently supervising one DST-sponsored project on human gait pattern generation for humanoid robot walk simulation. He is supervising one JRF, three Ph.D.s, six masters, and six undergraduate students. He joined MANIT Bhopal in 2019. He is currently an Assistant Professor with the Department of CSE, MANIT Bhopal, Bhopal, India. His research interests are machine learning, data science, deep learning, human activities recognition, analysis of biped locomotion, humanoid push recovery, natural language processing for text data, and biometric identification.



Youngshik Kim received the B.S. degree from Inha University, Incheon, South Korea, in 1996, and the M.S. and Ph.D. degrees from the University of Utah, Salt Lake City, UT, USA, in 2003 and 2008, respectively, all in mechanical engineering.

He is currently a Professor with the Department of Mechanical Engineering, Hanbat National University, Daejeon, South Korea. His main research interests include SMA actuator, bio-inspired robot, soft robot, compliant mobile robot, sensor fusion, motion control, and AI-based modeling and control.

Vishwanath Bijalwan (Senior Member, IEEE) was an Assistant Professor and the Head of the Department of ECE, Institute of Technology Gopeshwar, Chamoli, Uttarakhand, India (A Uttarakhand State Government Institute). He is a Postdoctoral Research Fellow with the Intelligent Control and Robotics Lab (ICRS), Hanbat National University, Daejeon, South Korea. He has over ten years of teaching experience and about a year of industrial experience. His current research interests include human gait, soft robotics, shape memory alloy actuators, human activity recognition, and deep learning.

Dr. Bijalwan is a Professional Member of ACM, a member of the Robotics and Automation Society and is actively engaged in IEEE R10 activities. He received the Excellence Award 2021 of the Uttarakhand State Government for Research.

Astha Verma received the bachelor's degree in computer science engineering from UPTU Lucknow, Lucknow, India, in 2016, and the master's degree from the Maulana Azad National Institute of Technology, Bhopal, India, in 2021.

She is currently working as a Software Engineer with Harbinger Systems Pvt. Ltd., Pune, India.

Ghanapriya Singh received the Ph.D. degree in signal processing from the Indian Institute of Technology (IIT) Delhi, New Delhi, India, in 2018.

She has been working as an Assistant Professor (Grade II) with the Department of Electronics Engineering, National Institute of Technology, Uttarakhand, Srinagar, India, since 2013. She has been working as an Assistant Professor (Grade I) with the Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra, Kurukshetra, India, since November 2022. During her research at IIT, she was working on a project with STMicroelectronics, Santa Clara, CA, USA. Her research work done at IIT was also showcased at CES, USA that is the largest consumer electronics show. She is an Inventor on five granted U.S. Patents and a European Patent. She is an Investigator in the project funded by Ministry of Electronics and Information Technology and Department of Science and Technology, Government of India. She has authored in high impact factor SCI indexed journals. Her current research interests include context awareness, image processing, speech processing, and signal processing for the Internet of Things (IoT).

Neha Gaud received the M.Phil. degree in computer science from the Institute of Computer Science, Vikram University, Ujjain. She is currently pursuing the Ph.D. degree in computer science with the School of Computer Science and Information Technology, Devi Ahilya Vishwavidyalaya, Indore, India.

She is having more than seven years of Vast Teaching Experience. She has published more than 10 publications in various SCI, Scopus Journal and refereed conferences. Her research includes human robot interaction, IoT and wearable sensor based health monitoring system, machine learning and AI.

Hangyeol Baek (Graduate Student Member, IEEE) received the B.S. degree in electronics engineering from Hanbat National University, Daejeon, South Korea, in 2022, where he is pursuing the master's degree in mechanical engineering.

His main research interests include SMA actuators, bio-inspired robots, soft robots, compliant mobile robots, and motion control.

Abdul Manan Khan (Senior Member, IEEE) received the Ph.D. degree in mechanical design engineering from Hanyang University, Seoul, South Korea, in 2016.

He is a Research Fellow at Hanbat National University, Daejeon, South Korea. His research interests include legged robots, reinforcement learning, deep learning, dynamics, control, and shape memory alloy-based actuators.

Dr. Khan is a member of the Robotics and Automation Society.