International Journal of Mathematical, Engineering and Management Sciences
Vol. 9, No. 4, 865-880, 2024
https://doi.org/10.33889/IJMEMS.2024.9.4.045

Ram Arti Publishers

Stable Biped Robot's Walk using Semi-Supervised ANN based Trajectory Generation within Yolov5 Algorithm based Identified Environment with Ditch

Seema Duhan
Department of Mathematics,
School of Engineering & Sciences, GD Goenka University, Sohna, 122103, Gurugram, Haryana, India.
E-mail: seemaduhan89@gmail.com

Ruchi Panwar
Department of Mathematics,
School of Engineering & Sciences, GD Goenka University, Sohna, 122103, Gurugram, Haryana, India.
Corresponding author: ruchipanwar321@gmail.com

(Received on October 25, 2023; Revised on December 6, 2023 & February 12, 2024 & April 2, 2024 & May 3, 2024; Accepted on May 9, 2024)

Abstract
The study introduces a stable walking pattern for a biped robot by employing a semi-supervised artificial neural network (ANN) to generate trajectories with a focus on reducing potential damage from small objects that are identified by Yolov5 algorithm. The ANN is utilized as a universal approximator to ensure smooth motion automatically by meeting predefined boundary conditions during its training. This trajectory generation approach is then compared with one another ANN- based method, with nonstop evaluations mainly focusing on position, velocity, and its acceleration profiles to maintain smooth motion. By analysis of trajectory derivatives and its curvature detects and auto corrects any discontinuities. Mathematical model created on from MATLAB 2023 and its simulations validate the trajectory's smoothness and demonstrating its effectiveness in enabling bipedal robots to navigate uneven terrain. The proposed method is very useful and more suitable for online adaptable trajectory generation by addressing collision avoidance and adaptability to various terrains, and overall stability in bipedal robot navigation comprehensively.

Keywords- Bipedal robot, Trajectory generation, Obstacle detection, Collision avoidance, Stability, Adaptability.

1. Introduction

Navigating uncertain and complex environments is indeed a tough task for a walking robot. Human can easily walk even in an unknown environment by using their subconscious knowledge to get the goal point where they want to go. So, humanoid robot is more suitable for unknown environment. To navigate uncertain environment, it is required that robot can avoid obstacles and can plan a collision free path. Walking in a complex unknown environment with obstacles and ditches, is very difficult task for a biped robot, because a small hit to robot by any obstacle can cause harm to the robot.

So, it is necessary to modify the trajectory and path as per the requirement of the environment. Several researchers (Huang et al., 2001; Shrivastava et al., 2007; Udai, 2008; Vandavilli et al., 2009; Farzadpour and Danesh, 2012; Kim, 2014; Xiaoguang and Ruyi, 2015) are there for obtaining trajectories for biped robot's locomotion. Trajectory planning is a critical aspect of locomotion that requires simultaneous tracking of different trajectories (Erbatur and Kurt, 2006; Aroche et al., 2011). Huang et al. (2001) used polynomials for hip and foot trajectory generation for a flat foot biped robot for uneven terrain. Panwar and Sukavanam (2019) used polynomial joint trajectory for legs and upper body and discussed the effect of various parameters on ZMP stable walk. For a more efficient approach rather than traditional methods, this study focuses on the application of Soft Computing techniques (Angeles-Garcia et al., 2022). Artificial neural networks (ANNs) offer approximate solutions for a wide range of systems (Parisi et al., 2003;

Watanabe and Johnson, 2018; Bhardwaj et al., 2023) due to its universal approximator property. Bhardwaj et al. (2023) proposed an ANN-based trajectory generating method for smooth walk on stairs. Numerous techniques and algorithms for obstacle identification and avoidance have been presented in the field of robot path planning in unknown environments (Desai et al., 2020; Alshammrei et al., 2022). Duguleana and Mogan (2016) suggested neural network-based reinforcement learning for obstacle avoidance for mobile robot. Alshammrei et al. (2022) proposed an optimal path planning algorithm, based on an improved Dijkstra algorithm typically operating in obstacle-free environments. To successfully traverse uncertain environment, robot need to possesses obstacle avoidance capability and the ability to plan collision-free path similar to human.

To successfully traverse uncertain environment, robot need to possesses obstacle avoidance capability and the ability to plan collision-free path similar to human. Several algorithms related to object detection and identification, based on the deep learning object methodology by Yang et al. (2020) and machine learning and deep learning approach are used by for Sharma and Mir (2020). The selection of the YOLO algorithm, give it distinct advantages in real-time object detection and precise identification within tough environment. Various YOLO algorithms are utilized in this domain for object detection (Yang et al., 2020; Sharma and Mir, 2020). These algorithms demonstrate robust autonomous learning capabilities, enabling accurate object localization and identification in complex and unknown environments (Jocher et al., 2021). A multi-target method based on YOLO has been proposed for object identification, offering an additional dimension to the capabilities of these algorithms (Francies et al., 2022). The benefits of YOLO algorithms in precise identification and real-time item recognition in complicated situations justify their cautious selection. Their quick processing fits very nicely with robotic systems' need for real-time operations, especially when it comes to handling obstacles and changing terrain. Furthermore, Neural Network (NN) algorithms' versatility across a variety of terrains emphasizes how well they respond to shifting environmental complexity by rapidly modifying and producing viable pathways.

The goal of the study is to present a simple trajectory generating method using ANN, designed specifically for a biped robot with five degrees of freedom (DOF). With this technology, step properties may be adjusted in real-time while walking, yielding quick results with low memory and computing requirements. In order to achieve flexibility during walking, the trajectory generating approach is constructed to accept differences in step characteristics. The ANN-generated trajectories follow stability requirements essential for bipedal locomotion through semi-supervised training, enabling effective and flexible trajectory planning.

1.1 Contribution Highlights
In order to identify obstacles in biped robots, this paper presents a novel approach that combines the Yolov5 vision-based architecture with unsupervised neural networks to create trajectories. The trajectory generating method using artificial neural networks (ANN) is contrasted with another method that is based on pre-established boundary constraints. Through the use of an unsupervised feed-forward neural network, the model is able to produce ankle, upper limb, and hip trajectories, which helps to formulate a walking pattern for biped humanoid robots. Evaluation criteria encompass ditch detection, ZMP stability assessment, and adaptable ankle trajectories, confirming the efficacy of the proposed method. Biped robots can recognize and negotiate ditches on uneven ground according to the technique. The paper's Sections 2 through 6 include details on the robot model's architecture and the suggested Artificial Neural network for its implementation.

2. Designing of Robot Model
Biped robot with 5 degrees of freedom with a flat foot is considered. Upper body (U), knee joint (K) and Hip joint (H) all these joints are revolute joints. $(l_1 + l_2)$ is the total length of the leg and l_3 and l_4 is the width of the foot respectively as given in **Table 1**. It is ensured that mass and the length of both legs are same as

shown in **Figure 1**. Walk of biped robot can be observed as the iteration of one step movement. Single support phase and double support phase are two stages of one cycle of walk. In SSP, robot's weight is on a stable foot while the swing foot is escalating in the air. In DSP, both feet are on the ground.

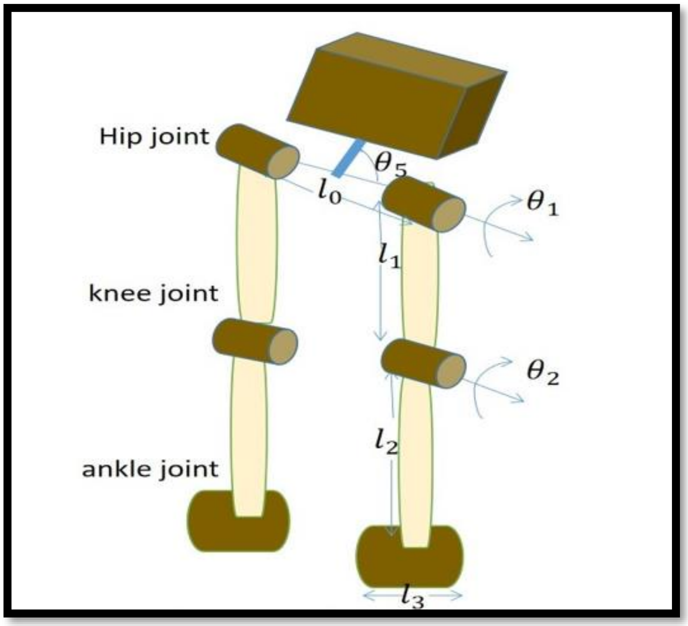


Figure 1. Schematic 3D biped model.

Table 1. Masses and lengths.

Link	Length	Value	Mass	Value
HK	l_1	14 units	M_1	4 units
KA	l_2	14 units	M_2	4 units
HU	l_5	16 units	M_6	65 units
HH	l_0	6 units	M_5	4 units
	l_3	4 units	M_3	0.4 units
	l_4	4 units	M_4	0.6 units

2.1 Function Approximation Property of Neural Network

A smooth function $n(s)$ from R^n to R^n is defined and $D_s \in R^n$ then for any $s \in D_s$ there exists some N number of hidden layer neurons and weights W and V as shown in Equation (1) such that

$$n(s) = W^T \sigma(V^T s) + \epsilon \quad (1)$$

where, ϵ is the ANN functional approximation error. In fact, for a positive number ϵ_N, one can find an ANN such that $\epsilon < \epsilon_N$ in D_s.

For a specified value of ϵ_N the ideal approximating ANN weights exist. Then, the estimation of $n(s)$ is

$$\hat{n}(s) = \hat{W}^T \sigma(\hat{V}^T s) \quad (2)$$

where, $\hat{W}$ and $\hat{V}$ are estimates of the ideal ANN weights that are used to provide some on-line weights tuning algorithms.

3. Walking Pattern Generation using Artificial Neural Network Approach

The walking motion of the biped can be determined by the hip, ankle and upper body trajectories which are given below in **Figure 2**.

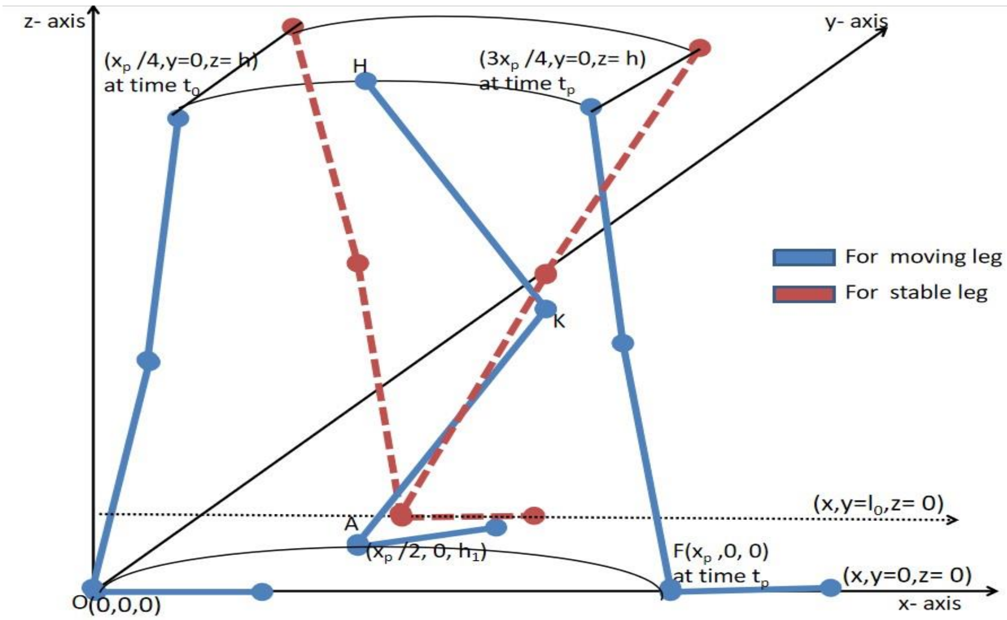


Figure 2. Walk planning.

3.1 Ankle Trajectory

During (t_o, t_p), the ankle joint A follows a trajectory from (0,0) to the final position (x_p, 0) in xz-plane as in **Figure 2**. Let ($x_A(t)$, 0, $z_A(t)$) is the coordinate of the ankle of swing leg. The trajectories for the ankle joint must satisfy the following position and velocity constraints at the starting and ending points.

For motion in x – direction

$$x_A(t_O) = x_0 ,\ x_A(t_p) = x_p ,\ \dot{x}_A(t_0) = 0,\ \dot{x}_A(t_p) = 0 \tag{3}$$

For motion in z – direction

$$z_A(t_O) = 0 ,\ z_A(t_p) = 0 ,\ \dot{z}_A(t_0) = h_1,\ \dot{z}_A(t_p) = 0 \tag{4}$$

where, x_p is step length and h_1 is step height which occurred at $x_m = x_p / 2$ position and $t_2 = t_p / 2$ time.

Using these conditions, the following 2 methods using neural network functions are proposed as in Equations (5) and (6) for x-trajectory.

Method-1

As an alternative, we have constructed a cubic polynomial for 4 boundary conditions and tried to find the coefficient of polynomial using neural networks. Each neural network has one coefficient as output and inputs are the value of all the boundary conditions. The trial coefficient using neural network have been constructed in such a way that the trajectory can satisfy the boundary conditions. Let $x_A(t)$ denotes the trajectory that can be written as in Equation (5),

$$x_A\,(t) = a_1 + a_2\,t + a_3 t^2 + a_4 t^3 \tag{5}$$

with coefficients,

$$a_1 = N_{11}\,(x_0, x_f, x_v, x_v, W_{11}, V_{11}),\ \ a_2 = N_{12}\,(x_0, x_f, x_v, x_v, W_{12}, V_{12}),$$
$$a_3 = N_{21}\,(x_0, x_f, x_v, x_v, W_{21}, V_{21})\ ,\ \ a_4 = N_{22}\,(x_0, x_f, x_v, x_v, W_{22}, V_{22}).$$

where, N_{11}, N_{12}, N_{21} and N_{22} are neural networks. Where, W_{11}, W_{12}, W_{21}, W_{22} and V_{11}, V_{12}, V_{21}, V_{22} are weights matrices from input to hidden and hidden to output layer respectively (Duhan and Panwar, 2023).

The neural networks N_1 and N_2 with input t and the weight matrices W_i (input to hidden layer) and V_i (hidden to output layer) are designed in such a way that N_1 vanish at t_0 and N_2 vanish at t_p. This ensures that trial function meets all boundary conditions if N_1 and N_2 satisfy the final and initial conditions, respectively; then final function generate a desired trajectory.

Method-2

$$x_A\,(t) = \left(\frac{t - t_0}{t_p - t_0}\right)^2\ N_1(t, W_1\,, V_1) + \left(\frac{t_p - t}{t_p\ -\ t_0}\right)^2 N_2(t, W_2\,, V_2\,) \tag{6}$$

where, the structure of N_1 and N_2 are similar to **Figure 3**.

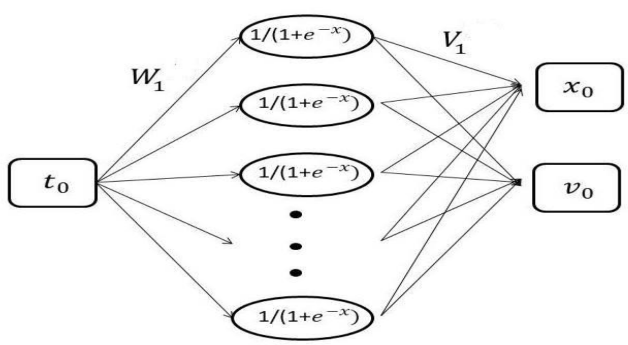


Figure 3. Methodology of N_1 in method-2 of trajectory generation.

The following neural network function is defined as proposed z-trajectory

$$z_A\,(t) = \left(\frac{x_m - x_A\,(t)}{x_m}\right)^2 \left(\left(\frac{x_p\ -\ x_A\,(t)}{x_p -\ x_o}\right) N_3\ \ (x_A\,(t), W_{3,}\, V_3\,) + \right.$$
$$\left.\left(\frac{x_A\,(t) - x_o}{x_p\ -\ x_o}\right) N_4\,(x_A\,(t), W_4\,, V_4\,)\right) \left(\frac{x_A\,(t)\ \ (x_p - x_A\,(t))}{(x_m\ \ (x_p - x_m))}\right) N_5\,(x_A\,(t), W_5\ \ , V_5) \tag{7}$$

The error function which has to be minimized by updating weights for x- trajectory in Equation (5) and (6) is given by:

$$E_1\ = (x_A\,(0) - x_0\,)^2 + \left(x_A\,(t_p) - x_p\,\right)^2 + \left(\frac{\partial x_A\,(0)}{\partial t}\right)^2 + \left(\frac{\partial x_A\,(t_p)}{\partial t}\right)^2 \tag{8}$$

The error function which has to be minimized by updating weights for z- trajectory in Equation (7) is given by:

$$E_2 = (z_A\ (0))^2 + \left(z_A\,(t_p)\right)^2 + (z_A\,(t_2) - h_1)^2 + \left(\frac{\partial z_A\,(t_2)}{\partial t}\right)^2 \tag{9}$$

We update the weights in E_1 and E_2 using gradient descent method.

3.2 Hip Trajectory

The hip follows a circular trajectory with center at stable leg's ankle joint A and radius ($l_1 + l_2$) along with the initial and final conditions. Boundary conditions related to hip coordinates ($x_H\,(t)$, $y_H\,(t)$, $z_H\,(t)$) are:

For motion in x-direction

$$x_H\,(t_0) = \frac{x_p}{4},\ x_H\,(t_p) = \frac{3x_p}{4},\ \dot{x_H}\,(t_0) = \upsilon_{H_1},\ \dot{x_H}\,(t_p) = \upsilon_{H_2} \tag{10}$$

For motion in y–direction

$$y_H\,(t_0) = y_s\,, y_H\,(t_2) = y_s + y_b\,, y_H\,(t_p) = y_s\,, \dot{y_H}\,(t_0) = \upsilon_{H_3}\,, \dot{y_H}\,(t_p) = \upsilon_{H_4}, \dot{y_H}\,(t_2) = 0 \tag{11}$$

For motion in z-direction

$$z_H\,(t_0) = h,\ z_H\,(t_p) = h,\ \dot{z_H}\,(t_0) = 0,\ \dot{z_H}\,(t_p) = 0 \tag{12}$$

where, h is maximum hip height at starting and final position, y_b is small displacement of hip in y-direction and y_s is initial position of hip in y-direction, υ_{H_1}, υ_{H_2} and υ_{H_3}, υ_{H_4} are initial and final velocity of hip in x and y-direction.

During the time interval (t_0, t_p), the hip trajectory in x and y-direction for 4 and 6 constraints respectively are computed by using the neural network functions. The trajectory in x-direction depends on t can be calculated by function in Equations (5) & (6) and weights are trained using error function E_3 given by,

$$E_3 = \left(x_H\,(t_0) - \frac{x_p}{4}\right)^2 + \left(x_H\,(t_p) - \frac{3x_p}{4}\right)^2 + \left(\frac{\partial x_H\,(t_0)}{\partial t} - \upsilon_{H_1}\right)^2 + \left(\frac{\partial x_H\,(t_p)}{\partial t} - \upsilon_{H_2}\right)^2 \tag{13}$$

Similarly, trajectory in y-direction depends on t for 6 conditions is

$$y_H(t) = x_A\,(t) = a_1 + a_2\,t + a_3 t^2 + a_4 t^3 + a_5 t^4 + a_6 t^5,$$

with coefficients

$a_1 = N_{31}\,(y_s, y_s + y_b, y_s, \upsilon_{H_3}, 0, \upsilon_{H_3}, W_{31}, V_{31}),$

$a_2 = N_{32}\,(y_s, y_s + y_b, y_s, \upsilon_{H_3}, 0, \upsilon_{H_3}, W_{32}, V_{32}),$

$a_3 = N_{33}\,(y_s, y_s + y_b, y_s, \upsilon_{H_3}, 0, \upsilon_{H_3}, W_{33}, V_{33}),$

$a_4 = N_{34}\,(y_s, y_s + y_b, y_s, \upsilon_{H_3}, 0, \upsilon_{H_3}, W_{34}, V_{34}),$

$a_5 = N_{35}\,(y_s, y_s + y_b, y_s, \upsilon_{H_3}, 0, \upsilon_{H_3}, W_{35}, V_{35}),$

$a_6 = N_{36}\,(y_s, y_s + y_b, y_s, \upsilon_{H_3}, 0, \upsilon_{H_3}, W_{36}, V_{36})$ (14)

Let error function E_4 given below is minimized by updating weights to satisfy initial, final and middle point conditions.

$$E_4 = (y_H(t_0) - y_s)^2 + \left(y_H(t_p) - y_s\right)^2 + y_H(t_2) - (y_b + y_s))^2 + \left(\frac{\partial y_H\,(t_0)}{\partial t} - \upsilon_{H_3}\right)^2 + \left(\frac{\partial y_H\,(t_2)}{\partial t}\right)^2 + \left(\frac{\partial y_H\,(t_p)}{\partial t} - \upsilon_{H_4}\right)^2 \tag{15}$$

The trajectory in z-direction is

$$z_H(t) = \sqrt{(l_1 + l_2)^2 - \left(x_H(t) - \frac{x_p}{2}\right)^2} \tag{16}$$

Kinematics solutions of these trajectories have been calculated using ANN based method (Panwar and Sukavanam, 2020).

In this paper, we focused exclusively on two dimensions: x and z.

3.3 Upper Body Trajectory
If upper body mass shift from one to another position on the frontal plane, then its suitable trajectory highly balanced the y-ZMP trajectory. Now it is considered that the upper body moves from middle of legs to the side of the stable foot in half time and then returns back to its location in half time. For the initial conditions,

$$U_R(0)=u_p,\ U_R(t_2)=u_m,\ U_R(t_p)=u_p\ \dot{U}_R(0)= v_{u1},\ \dot{U}_R(t_2)= v_{u2},\ \dot{U}_R(t_p)= v_{u1}.$$

The upper body trajectory in y-direction can also be generated using the Equation (14) given in section 3.2. The error function E_5 given below is minimized by updating weights to satisfy initial, final and middle point conditions.

$$E_5 = \left(y_H(t_0) - u_p\right)^2 + \left(y_H(t_p) - u_p\right)^2 + y_H(t_2) - u_m)^2 + \left(\frac{\partial y_H(t_0)}{\partial t} - v_{u1}\right)^2 + \left(\frac{\partial y_H(t_2)}{\partial t} - v_{u2}\right)^2 + \left(\frac{\partial y_H(t_p)}{\partial t} - v_{u1}\right)^2 \tag{17}$$

4. Stability Criteria and Path Planning
4.1 COM Trajectory for the Biped Robot
Center of mass (x_{com}, y_{com}, z_{com}) for the biped robot can be calculated by the formula,

$$x_{com} = \frac{\sum x_k m_k}{m_k},\ y_{com} = \frac{\sum y_k m_k}{m_k},\ z_{com} = \frac{\sum z_k m_k}{m_k}.$$

where, (x_k, y_k, z_k) co-ordinates of center of mass of each link k and m_k is mass of each link k.

4.2 ZMP Stability Criteria
ZMP is the point where the sum of all moments along the axis parallel to the ground equal to zero. If the ZMP is inside the supported region, then the biped robot can walk in a stable manner. The value of x-ZMP and y-ZMP can be calculated by,

$$xzmp = \frac{\sum_{k=1}^{n} m_k\left(x_k(\ddot{z}_k + g) - \ddot{x}_k z_k\right)}{\sum_{k=1}^{n} m_k\left(x_k(\ddot{z}_k + g)\right)} \tag{18}$$

$$yzmp = \frac{\sum_{k=1}^{n} m_k\left(y_k(\ddot{z}_k + g) - \ddot{y}_k z_k\right)}{\sum_{k=1}^{n} m_k\left(y_k(\ddot{z}_k + g)\right)} \tag{19}$$

where, the mass of link k is m_k, the number of links are n and gravity is g.

4.3 YOLO Algorithm: Real Time Objects Detection
Yolo algorithms are used as object detectors for images. YOLO has undergone several iterations with Yolov2, Yolov3, Yolov5 and Yolov8 introducing improvements in terms of accuracy and speed. These are the best model approaches, for detecting a ditch. Yolov5 introduces a modified architecture compared to its predecessors. It includes a large number of layers and parameters, allowing the model to capture more

complex features and patterns in the data. Yolov5 outperforms earlier other objects detection models in terms of accuracy and speed. Yolov5 demonstrates versatility. So, in this paper for detecting the ditches, Yolov5 is used and results are satisfactory. The performance of Yolov5 in detection of ditches is outstanding that fulfill the requirement of current work.

5. Result

For simulation of the trajectories, parameters used are given in **Table 2**. Initially, swing foot and stable foot are on the ground and they are positioned at $0 < x < 5$, $2 < y < 2$ and $7 < x < 12$, $4 < y < 8$ respectively.

Table 2. Parameters.

Gait Analysis	**Parameter**	**Value**
Step parameters	Step length (x_p) Step height (h_1) Initial time (t_0) Initial position (x_0) Total time (t_p)	12 units 3 units 0 sec 0 units 2 sec
Hip	Position of stable leg ($y_s = l_0$) Position of swing leg Displacement (y_b)	6 units 0 units 5 units
Upper body	Initial position (y_l) Final position(y_a)	3 units 7 units
Learning rate (α)		0.0001
Hidden layer neurons in one ANN function	For 4 boundary conditions For 6 boundary conditions	6 10
Convergence time	For 4 boundary conditions For 6 boundary conditions	3.73 sec 75 Sec

5.1 Comparative Analysis of both Methods for Ankle Trajectory Planning

Using given parameters, **Figures 4** present the x-trajectory of ankle joint for the given boundary conditions using different methods of neural network. **Figure 4(A)** compared the results of method-1 and 2 in MATLAB to ensure that proposed methods have generated the ankle trajectory in acceptable manner. As we can observe from **Figures 4(A)** and **4(B)** that all the trajectories and their derivative trajectories are continuous and satisfy the given boundary conditions. Note that the trajectory generated using method-1 and method -2 are almost the same but acceleration trajectory is smoother for method 2 which is required for ankle trajectory generation. For 3 different sets of data in method-2, **Figures 5(A)**, **5(B)** and **5(C)** show the ankle trajectory x_A dependence on boundary neural networks N_1 and N_2. The computed solutions at some interpolation points for method-1 and method-2 are given in **Table 3**. So, it can be concluded that both methods generate smooth trajectory but convergence time is lesser for method-1 although method-2 is smoother and more adaptable which is best suited for ankle trajectory as in **Figure 6**.

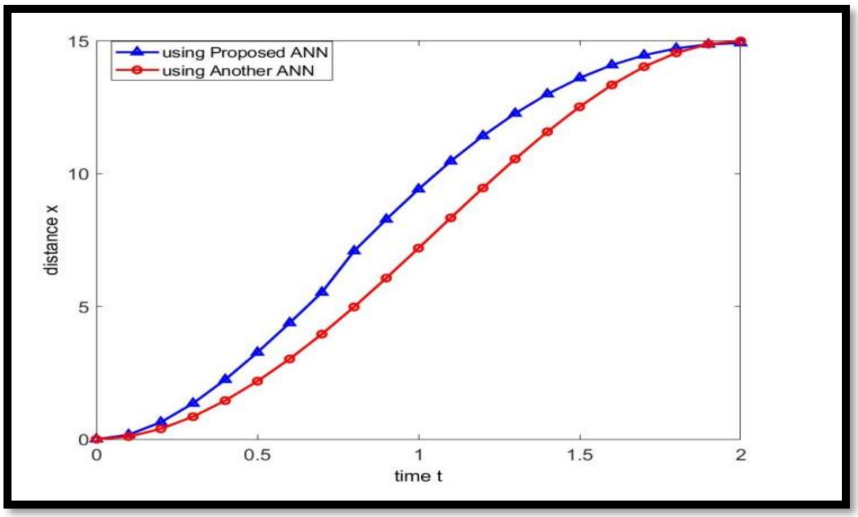
Error for method-1 and method-2 converges smoothly and continuously up to desired accuracy for both methods in **Figure 7**.

(i) Trajectory in all the cases is differentiable, continuous, and satisfy all the boundary conditions.
(ii) These methods are applicable for online trajectory generation.
(iii) It has been observed that method-2 is more suitable for online trajectory generation due to accuracy but method-1 takes less computational time.

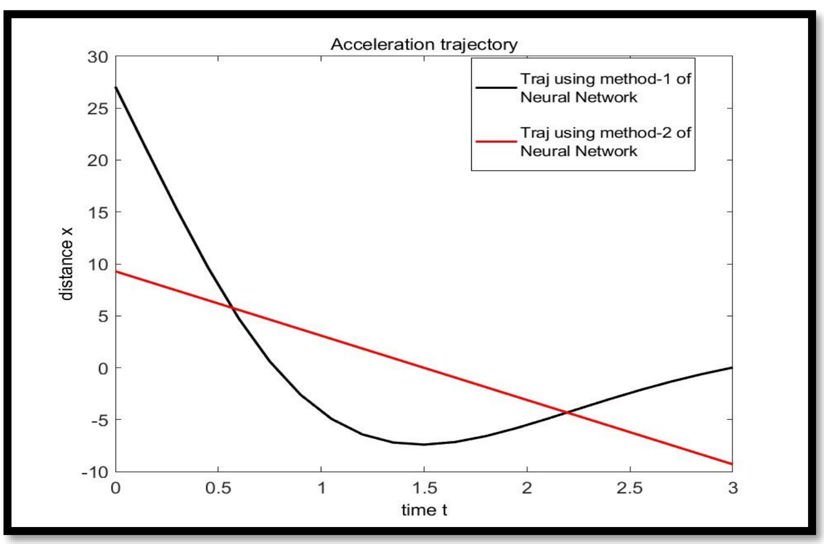
So, it can conclude that method-2 can be used for adaptable online trajectory generation while method-1 can be used where trajectory has to plan for many constraints but frequent modification in trajectory is not required.

Table 3. Simulation based observations.

Parameter's Desired Value (x_o, x_f, x_{v1}, x_{v2})	Parameter's Actual value (x_{A_i} (t_o), x_{A_i} (t_f), x_{A_i} (t_o), x_{A_i} (t_f))	Time T	Error E
Method -1			
(0, 14, 0, 0)	(0, 13.9828, 0, 0.0265)	3.73s	0.0001
(4, 24, 5, 5)	(4.0039, 23.9926, 5, 5.0264)	3.93s	0.0001
(4, 24, 2, 8)	(4.0039, 23.9926, 2, 8.0264)	3.73s	0.0001
Method -2			
(0, 14, 0, 0)	(0.005, 13.9984, 0.008, 0.0017)	2.41s	0.0001
(4, 24, 5, 5)	(3.9958, 24.0017, 4.9913, 4.9982)	1.83s	0.0001
(4, 24, 2, 8)	(4.0039, 23.9926, 2.0089, 8.0018)	1.67s	0.0001

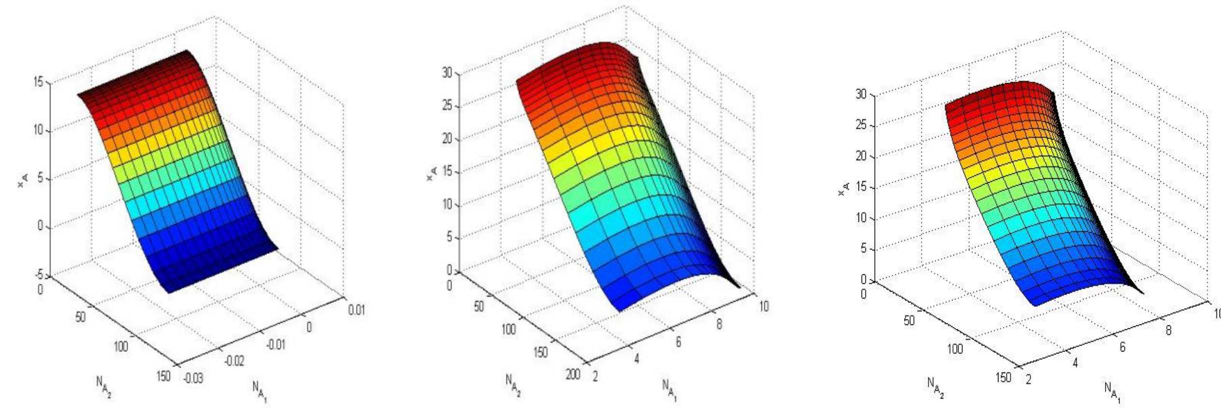


(a) Generated x-t joint trajectories.



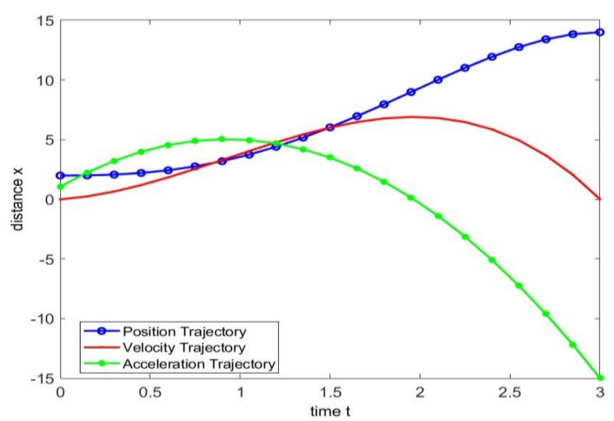
(b) Acceleration trajectory.

Figure 4. Trajectories generation for ankle joint using both neural network methods.

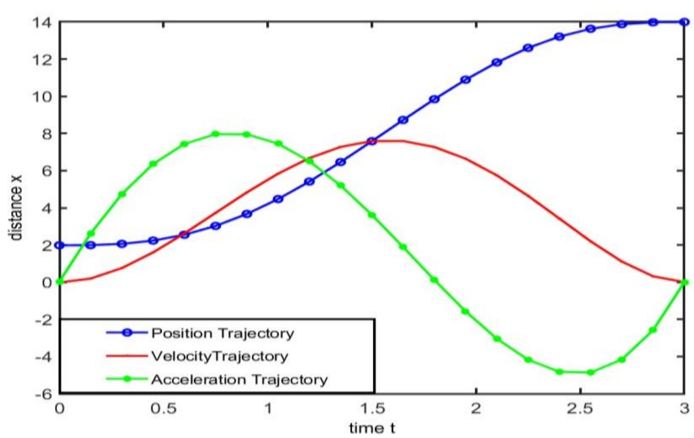


(a) $x_f = 14$, $x_0 = 0$, $x_{v1} = 0$, $x_{v2} = 0$. (b) $x_f = 24$, $x_0 = 4$, $x_{v1} = 5$, $x_{v2} = 5$. (c) $x_f = 24$, $x_0 = 4$, $x_{v1} = 2$, x_{v2}=8.

Figure 5. X_A dependance on N_1 and N_2 in proposed method.



(a) Generated trajectories for 4 constraints.



(b) Generated trajectories for 6 constraints.

Figure 6. Trajectories generation for different constraints using proposed neural network method.

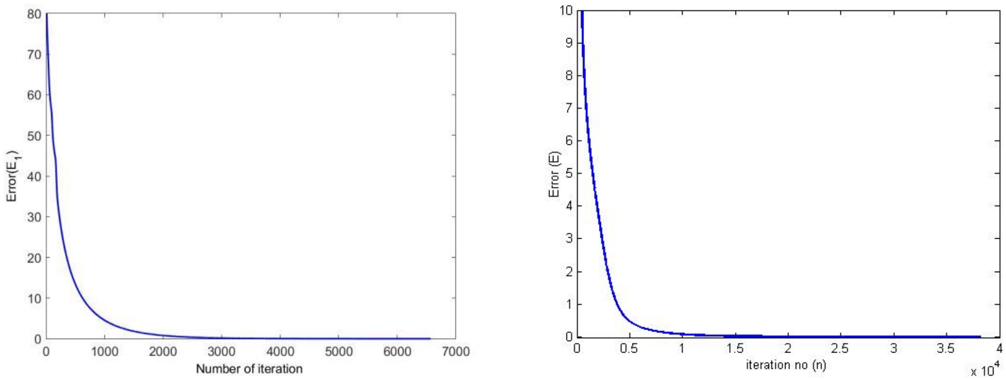
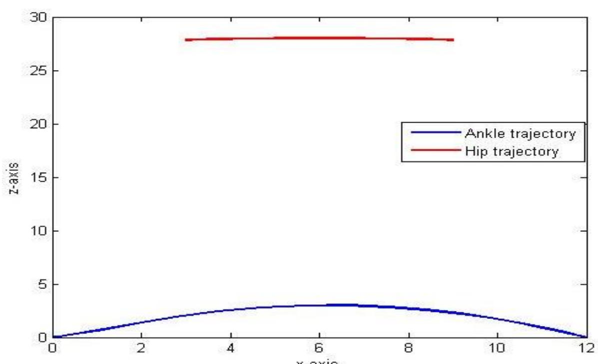


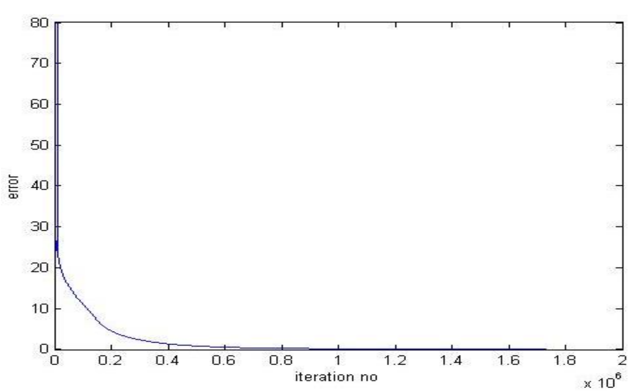
Figure 7. Error for method 1 and 2 for x- ankle trajectory generation.

5.2 Performance Evaluation of Desired Trajectory Generation Methods

After this concluded remark, ANN in both the methods can generate trajectories that facilitate smooth and natural movements by ensuring that the motion transitions seamlessly between different stances or poses. These methods help in avoiding abrupt changes in motion that could affect the robot's stability or efficiency while walking on uneven terrain or navigating obstacles. Using these methods, the biped robot will not have to face jerk issue during control.



(a) Generated joint trajectories.



(b) Errors in generating hip trajectory using nn.

Figure 8. Trajectories generation for ankle joint using neural network.

It is observed from **Figure 6** that ankle position and acceleration trajectories using method-2 of NN are more continuous, differentiable, smooth and also satisfy all the boundary conditions. So, in this paper, method-2 is used for ankle trajectory generation and method-1 is used for other trajectories.

Figure 8 Presents the desired x versus z using ANN for ankle and hip joints. The **Figure 8** another half shows that errors converge smoothly upto desired accuracy in approximating the boundary conditions for trajectory generation in generating hip trajectory.

The upper body moves in y-direction for balance. So, the trajectory for the upper body in y-direction depending on time t for two steps is shown in **Figure 9**.

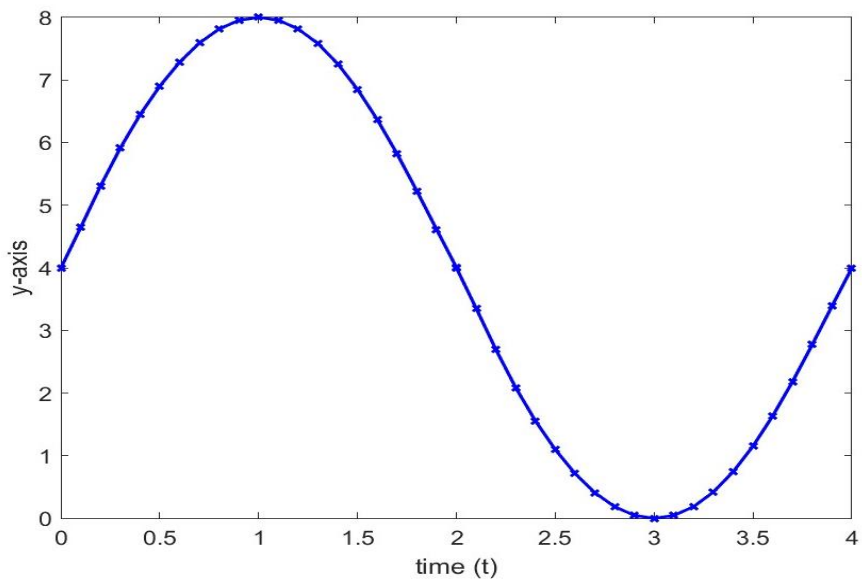
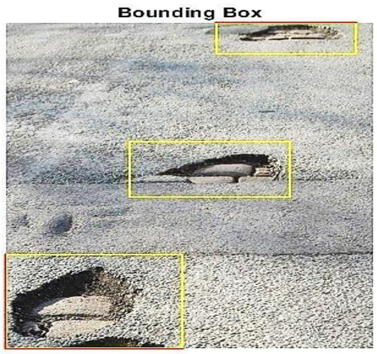


Figure 9. y-t upper body trajectory using neural networks.

5.3 Performance Metrics by Robotic Navigation Performance

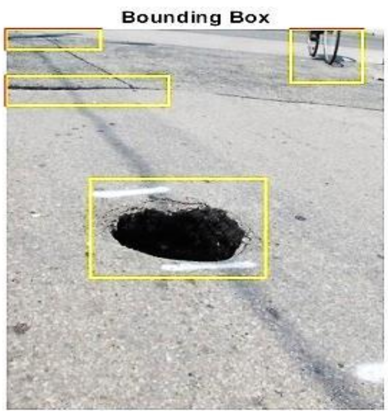
5.3.1 Exploration of Unknown Environment and Obstacle Avoidance

With the proposed approach, the robot is capable to detect and avoid or cross the obstacles in its path effectively. The generation of a comprehensive map from the object detection phase displayed a detailed spatial distribution of identified objects. The implemented trajectory planning methodology successfully guided the biped robot in real-time scenarios within the unknown environment. Real-time execution demonstrated the robot's ability to adapt the change dynamically and navigate safely. In **Figure 10**, ditches are identified using YOLO algorithm. Now after identification of complex environment, the joint trajectories can be planned accordingly. Using planned ANN joint trajectories, this biped robot can cross a ditch/obstacle in path by modifying the trajectory online accordingly as in **Figure 11** with ZMP stability.

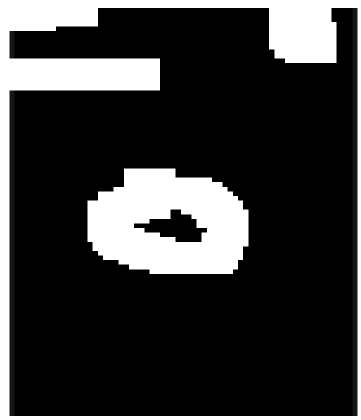


(a)

(b)

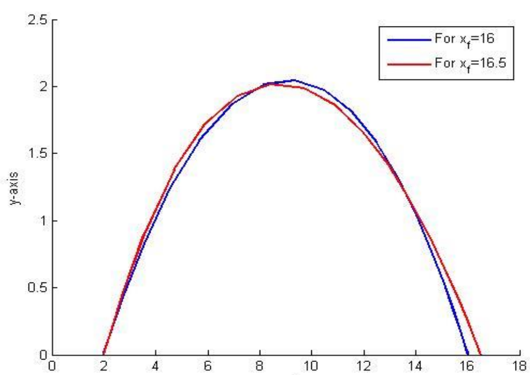


(c)



(d)

Figure 10. Ditch identification using Yolo algorithm.



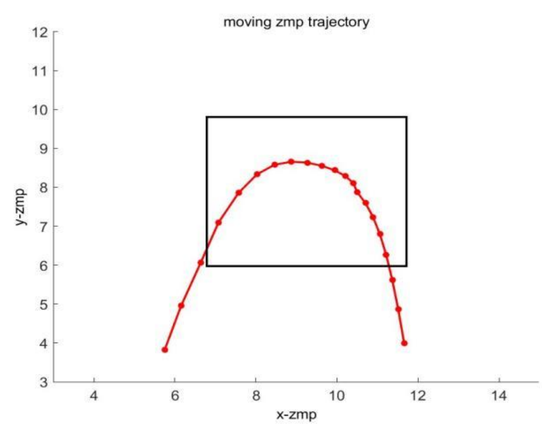


Figure 11. Illustration of a changed ankle trajectory to cross a ditch, with accompanying Zero Moment Point (ZMP) calculations for the altered path.

5.3.2 Stability for these Trajectories

Our biped robot can walk on planned uneven surface easily with ZMP stability as in **Figures 11**. x- and y-ZMP for 2 steps of this biped robot have been calculated and plotted in given **Figure 12** with the black borders lines of the supporting polygon. A COM trajectory of whole body is shown in **Figure 12**. It can be observed from **Figure 11** and **Figure 12** that the ZMP trajectory is inside the support polygon for the proposed joint trajectories and can adapt a change in ankle trajectory to cross a ditch.

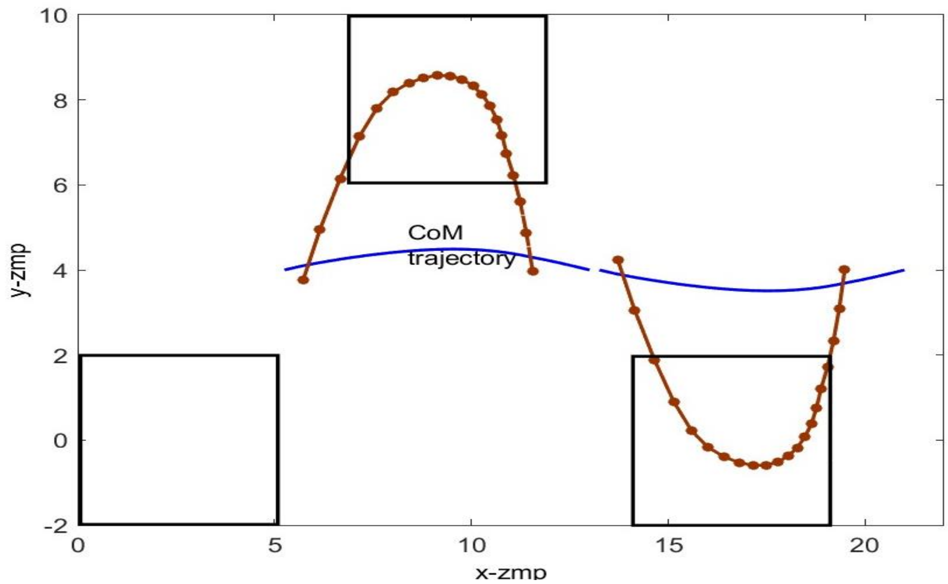


Figure 12. x versus y-ZMP and COM with in the support polygon for a sequence of 2 steps.

6. Conclusion and Future Work

An Artificial Neural Networks (ANN) guided trajectory planning system has shown exceptional flexibility across various landscapes. It effectively adapts and generates optimal paths, navigating through dynamic environmental challenges with ease. The trajectories generated by the artificial neural network are continuous and meet boundary conditions throughout training, resulting in smooth motion for the biped robot without jerks. The Zero Moment Point (ZMP) trajectory maintains stability on various surfaces by staying within the support polygon. Consequently, the biped robot demonstrates stable walking across different terrains. Our ANN-based path planning approach efficiently balances computational speed, trajectory quality, and adaptability to diverse scenarios. The study also introduces a vision-based obstacle detection method using the Yolov5 algorithm, effectively reducing collision risks. Looking ahead, future research avenues include further refining and optimizing the NN-based trajectory planning algorithm, validating the proposed approach in real-world scenarios, exploring additional enhancements for complex environments, and integrating real-time feedback mechanisms to improve adaptability and responsiveness during navigation.

6.1 Limitations

6.1.1 Sensitivity to Environmental Variations

The efficacy of the suggested approach may be impacted by changes in the surrounding environment, such as the trajectory planning system easily handles unpredictable changes in the terrain. This sensitivity may have an impact on how well obstacles are detected, particularly in complex environments.

6.1.2 Model Complexity and Training Overhead
With regards to show intricacy and preparing above, involving unaided feed-forward brain networks for direction advancement could give challenges. Having long-term commitments could be really important for getting things done and preparing of these organizations since they might call for complex model plans and critical figuring assets.

6.2 Future Scopes
Exploring enhanced stability in bipedal robot locomotion through semi-supervised Artificial Neural Network trajectory generation within a Yolov5 algorithmic framework for environment recognition, particularly focusing on navigating challenging terrain features like ditches. This research aims to improve the robot's adaptability and robustness in real-world scenarios, advancing bipedal robotics applications.

Conflict of Interest
All authors declare that they have no conflicts of interest.

Acknowledgments
This study did not receive dedicated funding from public, commercial, or non-profit entities. The authors express gratitude to the editor and anonymous reviewers for their insightful feedback, contributing to the enhancement of this research.

References
Alshammrei, S., Boubaker, S., & Kolsi, L. (2022). Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance. *Computers, Materials & Continua*, *72*, 5939-5954.

Angeles-García, Y., Calvo, H., Sossa, H., & Anzueto-Ríos, Á. (2022). Dynamic balance of a bipedal robot using neural network training with simulated annealing. *Frontiers in Neurorobotics*, *16*, 934109. https://doi.org/10.3389/fnbot.2022.934109.

Aroche, O.N., Cozatl, E.R., & Jimenez, F.C. (2011). Kinematic analysis and computation of ZMP for a 12-internal-dof biped robot. In *13th World Congress in Mechanism and Machine Science* (pp. 19-25). Guanajuato, Mexico.

Bhardwaj, G., Mishra, U.A., Sukavanam, N., & Balasubramanian, R. (2023). Neural network temporal quantized Lagrange dynamics with cycloidal trajectory for a toe-foot bipedal robot to climb stairs. *Applied Intelligence*, *53*(9), 10995-11018. https://doi.org/10.1007/s10489-022-03921-6.

Desai, K., Parikh, S., Patel, K., Bide, P., & Ghane, S. (2020). Survey of object detection algorithms and techniques. In *Cybernetics, Cognition and Machine Learning Applications: Proceedings of ICCCMLA 2019* (pp. 247-257). Springer, Singapore.

Duguleana, M., & Mogan, G. (2016). Neural networks based reinforcement learning for mobile robots obstacle avoidance. *Expert Systems with Applications*, *62*, 104-115.

Duhan, S., & Panwar, R. (2023). Online adaptable trajectory using artificial neural networks for unknown environment. *Procedia Computer Science*, *230*, 377-387.

Erbatur, K., & Kurt, O. (2006). Humanoid walking robot control with natural ZMP references. In *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics* (pp. 4100-4106). IEEE. Paris, France.

Farzadpour, F., & Danesh, M. (2012). GA based trajectory generation for a 7-DOF biped robot by considering feet rotation during double support phase. *Advanced Materials Research*, *463*, 1252-1255.

Francies, M.L., Ata, M.M., & Mohamed, M.A. (2022). A robust multiclass 3D object recognition based on modern YOLO deep learning algorithms. *Concurrency and Computation: Practice and Experience*, *34*(1), e6517.

Huang, Q., Yokoi, K., Kajita, S., Kaneko, K., Arai, H., Koyachi, N., & Tanie, K. (2001). Planning walking patterns for a biped robot. *IEEE Transactions on Robotics and Automation*, *17*(3), 280-289.

Jocher, G., Stoken, A., Borovec, J., Chaurasia, A., Changyu, L., Hogan, A., Hajek, J., Diaconu, L., Kwon, Y., Defretin, Y. and Lohia, A., & Ingham, F. (2021). ultralytics/yolov5: v5. 0-Yolov5-P6 1280 models, AWS, Supervise. ly and YouTube integrations. *Zenodo*.

Kim, J.W. (2014). Online joint trajectory generation of human-like biped walking. *International Journal of Advanced Robotic Systems*, *11*(2), 19. https://doi.org/10.5772/57415.

Panwar, R., & Sukavanam, N. (2019). Effect of upper body motion on biped robot stability. In: Deep, K., Jain, M., Salhi, S. (eds) *Decision Science in Action. Asset Analytics* (pp. 237-250). Springer, Singapore. https://doi.org/10.1007/978-981-13-0860-4_18.

Panwar, R., & Sukavanam, N. (2020). Trajectory tracking using artificial neural network for stable human-like gait with upper body motion. *Neural Computing and Applications*, *32*(7), 2601-2619.

Parisi, D.R., Mariani, M.C., & Laborde, M.A. (2003). Solving differential equations with unsupervised neural networks. *Chemical Engineering and Processing: Process Intensification*, *42*(8-9), 715-721.

Sharma, V.K., & Mir, R.N. (2020). A comprehensive and systematic look up into deep learning based object detection techniques: A review. *Computer Science Review*, *38*, 100301.

Shrivastava, M., Dutta, A., & Saxena, A. (2007). Trajectory generation using GA for an 8 DOF biped robot with deformation at the sole of the foot. *Journal of Intelligent and Robotic Systems*, *49*, 67-84.

Udai, A.D. (2008). Optimum hip trajectory generation of a biped robot during single support phase using genetic algorithm. In *2008 First International Conference on Emerging Trends in Engineering and Technology* (pp. 739-744). IEEE. Nagpur, India.

Vundavilli, P.R., & Pratihar, D.K. (2009). Soft computing-based gait planners for a dynamically balanced biped robot negotiating sloping surfaces. *Applied Soft Computing*, *9*(1), 191-208.

Watanabe, T., & Johnson, E.N. (2018). Trajectory generation using deep neural network. In *2018 AIAA Information Systems-AIAA Infotech@ Aerospace* (p. 1893).

Xiaoguang, Z., & Ruyi, H. (2015). Research on humanoid robot slope gait planning. *The Open Automation and Control Systems Journal*, *7*(1), 1002-1009.

Yang, J., Li, S., Wang, Z., Dong, H., Wang, J., & Tang, S. (2020). Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges. *Materials*, *13*(24), 5755.

Publisher's Note- Ram Arti Publishers remains neutral regarding jurisdictional claims in published maps and institutional affiliations.