# Gait Control of 6DoF Biped Robot Using RL-Based Controller

Gokul G I
*Department of Electrical Engineering*
*College of Engineering Trivandrum*
Kerala, India
gokul20gi00@gmail.com

Arlene Davidson R
*Department of Electrical Engineering*
*College of Engineering Trivandrum*
Kerala, India
arlene@cet.ac.in

**doi**: https://doi.org/10.37285/bsp.SACAD2025.12

***Abstract*—This paper presents two distinct gait control strategies for a 6DoF biped robot within the Robot Operating System (ROS) framework: a cubic spline method with Proportional-Integral-Derivative (PID) control for smooth trajectory generation and joint tracking, and a Reinforcement Learning (RL) approach using the Deep Deterministic Policy Gradient (DDPG) algorithm for adaptive gait optimization. The kinematic model of the 6DoF biped robot is developed using Denavit-Hartenberg (DH) parameters, with the design validated in Fusion 360. The cubic spline method ensures continuous joint trajectories, regulated by PID controllers, while RL optimizes stride length and foot placement. Simulations in Gazebo demonstrate precise angle tracking for the PID-based approach and stable locomotion for the RL model on planar terrain. These methods enhance adaptability for applications in elderly care, rehabilitation, and disaster response.**

***Index Terms*—Biped Robot, Gait Control, Cubic Spline, PID Control, Reinforcement Learning, ROS**

## I. INTRODUCTION

Bipedal robots, designed to emulate human locomotion, are pivotal for navigating human-centric environments in applications such as healthcare, search-andrescue, and space exploration [1]. However, achieving stable and efficient gait remains a significant challenge due to the complex dynamics and environmental variability inherent in bipedal locomotion. Traditional model-based methods, while effective in controlled settings, often lack the adaptability required for diverse real-world conditions, necessitating advanced approaches like Reinforcement Learning (RL) [29]. This study addresses these challenges by implementing two independent gait control strategies within the Robot Operating System (ROS) framework: (1) a cubic spline method with PID control for smooth trajectory generation and precise joint tracking, and (2) an RL-based approach for adaptive gait optimization. The objectives are to optimize gait parameters—stride length, frequency, and foot placement—for human-like walking and to ensure robustness across varied terrains. A kinematic model for a 6-DoF robot is developed, with performance validated through simulations in Gazebo. The field of bipedal locomotion has been extensively studied, with control approaches broadly categorized into model-based, learning-based, and hybrid methods.

Model-based methods rely on precise kinematic and dynamic modeling. For instance, proposed a variable horizon Model Predictive Control (MPC) [10] approach focusing on swing foot dynamics to enhance stability, while [11] utilized whole-body MPC for torque-controlled humanoids, computing real-time feedback for balance. The compass gait model, reviewed by [15], simplifies dynamics by modeling legs as rigid hinges, aiding controller design [12]. Additionally, [18] developed a universal walking framework using dynamic models and quadratic programming, offering adaptability across conditions.

Reinforcement Learning has gained significant traction for its adaptability in dynamic environments. Integrated gait principles with RL [29], using adversarial critics to adapt to changing conditions, while [2] analyzed deep RL policies, proposing solutions for stability issues. Applied the Deep Deterministic Policy Gradient (DDPG) [17] algorithm for bipedal walking, demonstrating robustness in simulation. Furthermore, [21] achieved real-world humanoid locomotion using RL, underscoring its practical potential. Hybrid approaches, combining traditional and learningbased methods, have also shown promise. proposed a Center of Mass (CoM) [16] trajectory control system for stable walking by dynamically adjusting foot placement, while [20] designed anthropomorphic gaits to mimic human patterns for smoother transitions. Adapted quadrupedal control principles [14] to enhance humanoid robustness.

Kinematic and dynamic studies provide foundational insights for these control strategies. The modelling of lower body kinematics [24] [5], emphasizing joint articulation, while [23] applied RL for precise footstep planning on complex terrains. [13] introduced active toe joints to enhance stability, and recent advances by [19] and [22] highlight the importance of sensor integration and kinematic modeling of humanoid torsos. These studies collectively underscore the need for adaptive and robust control strategies, motivating the exploration of both model-based and RL-based approaches in this work. By combining these strategies within the ROS framework, this study aims to advance the development of stable, efficient, and adaptable bipedal locomotion for real-world applications.

## II. METHODOLOGY

The methodology involves designing and implementing two gait control strategies for a 6DoF biped robot within the ROS framework, with validation through simulation. The process includes kinematic modeling, mechanical design, controller implementation, and performance evaluation.

### *A. Robot Design and Modeling*

The biped robot is designed with six degrees of freedom (DoF) per leg, comprising hip, knee, and ankle joints, all configured as revolute joints in the sagittal plane. The mechanical structure is modeled in Fusion 360, specifying components such as the waist, thighs, shanks, and feet. The kinematic model is developed using

Denavit-Hartenberg (DH) parameters, implemented in ROS via Unified Robot Description Format (URDF) files for simulation in Gazebo and visualization in RViz.

*B. Control Strategies*

Two control strategies are implemented:

1) Cubic Spline with PID Control: This method generates smooth joint trajectories using cubic splines, with PID controllers ensuring precise tracking of joint angles. The trajectories are computed in Python nodes within ROS, and PID gains are tuned using the RQT GUI.
2) Reinforcement Learning with DDPG: The Deep Deterministic Policy Gradient (DDPG) algorithm optimizes gait parameters by learning from simulated interactions in Gazebo. The state space includes joint angles, velocities, and Center of Mass (CoM) position, while actions consist of joint torques or position commands.

*C. Simulation Environment*

Simulations are conducted in Gazebo, integrated with ROS, to evaluate both controllers. The cubic spline method is tested for angle tracking accuracy, while the RL approach is assessed for gait stability and adaptability. MATLAB is used for analyzing joint angle trajectories, and TensorBoard monitors RL training progress.

*D. Performance Metrics*

Performance is evaluated based on:

- Tracking Accuracy: Root Mean Square Error (RMSE) between desired and actual joint angles for the PID-based controller.
- Gait Stability: CoM stability and foot placement accuracy for both controllers.
- Reward Convergence: Average reward trends during RL training to assess learning efficiency.

## III. MODELING

*A. Kinematic Modeling*

The 6DoF biped robot is modeled using DenavitHartenberg (DH) parameters to define the kinematic chain. Each leg consists of three revolute joints (hip, knee, ankle), enabling motion in the sagittal plane. The matrix representing the transformation of individual joint is given by:

$$T_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The DH parameters are listed in Table I. Inverse kinematics [13] compute joint angles for desired foot positions. The equations ensure precise foot placement by solving for $\theta_0$, $\theta_1$, and $\theta_2$ based on the end-effector coordinates. The MoveIt package in ROS validates joint ranges and workspace reachability, ensuring the kinematic model aligns with physical constraints.

TABLE I: DH Parameters for 6DoF Biped Robot

| Joint | $\theta_i$ | $d_i$ | $a_i$ | $\alpha_i$ |
|---|---|---|---|---|
| 0 | $\theta_0$ | 0 | 0 | 0 |
| 1 | $\theta_1$ | 0 | 0 | 0 |
| 2 | $\theta_2$ | 0 | 0 | $L_1$ |
| 3 | $\theta_3$ | $L_3$ | 0 | 0 |
| 4 | $\theta_4$ | 0 | 0 | $-L_2$ |
| 5 | $\theta_5$ | 0 | 0 | $-L_1$ |

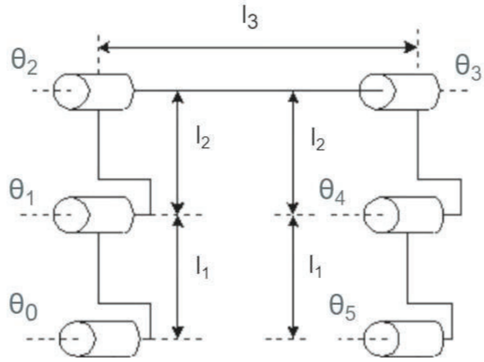


Fig. 1: DH Parameter Visualization of 6DoF Biped Robot

Figure 1 illustrates the kinematic chain with DH parameters, showing the coordinate frames for each joint. This visualization aids in understanding the robot's structure and motion planning.

*B. Design in Fusion 360*

The robot's mechanical design is developed in Fusion 360, with components including the waist (0.1 m wide), thighs (0.3 m long), shanks (0.3 m long), and feet (0.15 m long, 0.08 m wide). Each component is modeled with precise dimensions and assembled with revolute joints. Simulations in Fusion 360 test joint motion ranges, ensuring no collisions occur during gait cycles. The URDF model is exported to ROS, and RViz visualizes the joint connections, confirming structural integrity.

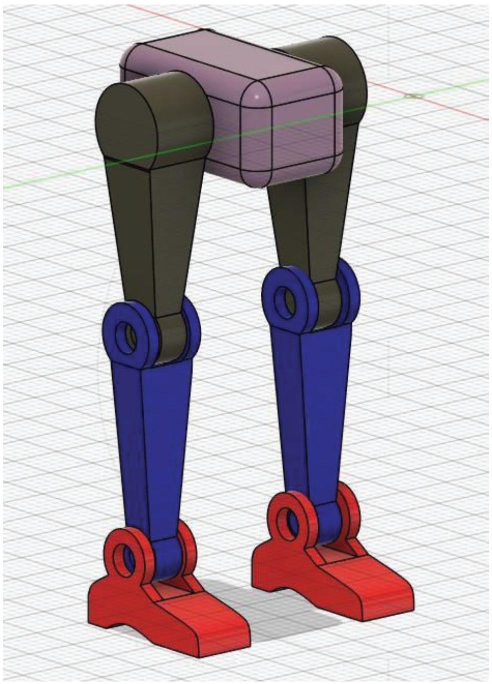

Fig. 2: Fusion 360 Model of 6DoF Biped Robot

Figure 2 shows the assembled robot model in Fusion 360, highlighting the articulated joints and component alignment.

## IV. CONTROLLER DESIGN

### *A. Cubic Spline with PID Control*

The cubic spline method generates smooth joint trajectories to ensure continuity in position, velocity, and acceleration:

$$\theta_i(t) = a_i(t-t_i)^3 + b_i(t-t_i)^2 + c_i(t-t_i) + d_i \quad (2)$$

The coefficients ($a_i$, $b_i$, $c_i$, $d_i$) are computed to satisfy boundary conditions at keyframes, ensuring smooth transitions. Trajectories are implemented in ROS Python nodes, driving actuators via PID controllers:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (3)$$

PID gains (Table II) are tuned using ROS's RQT GUI to minimize tracking errors. The feedback loop compares desired and actual joint angles, adjusting control signals for precision. The tuning process involves iterative adjustments to achieve minimal overshoot and steady-state error.

TABLE II: PID Gains for Joint Angles

| Joint | Gain |
|---|---|
| Hip $k_p$ | 60 |
| Hip $k_i$ | 10 |
| Hip $k_d$ | 20 |
| Knee $k_p$ | 60 |
| Knee $k_i$ | 5 |
| Knee $k_d$ | 10 |
| Ankle $k_p$ | 20 |
| Ankle $k_i$ | 4 |
| Ankle $k_d$ | 8 |

### *B. Reinforcement Learning Controller*

The DDPG algorithm, which works well in continuous action spaces, is used by the RL controller. stability, forward motion, and energy efficiency are all balanced in the reward function's design:

$$r = b_1 v_x - b_2 \dot{y}^2 - b_3 \dot{z}^2 - b_4(b_5\theta^2 + b_6\phi^2 + b_7\psi^2) - b_8|\dot{z}| - b_9 \sum a^2 \quad (4)$$

The coefficients ($b_1$ to $b_9$) are tuned to prioritize forward velocity ($v_x$) while penalizing lateral and vertical oscillations ($\dot{y}$, $\dot{z}$), orientation errors ($\theta$, $\phi$, $\psi$), and excessive actuator effort ($\sum a^2$). The state space includes joint angles, velocities, CoM position, and foot contact states, while actions are joint torques or position commands. Training occurs in Gazebo using StableBaselines3, with domain randomization (e.g., varying terrain friction) enhancing robustness. Hyperparameters, such as learning rate and discount factor, are fine-tuned for convergence, monitored via TensorBoard.

TABLE III: Reward function parameters

| Parameter | Value |
|---|---|
| $b_1$ (Forward velocity) | 100.0 |
| $b_2$ (Lateral velocity) | 10.0 |
| $b_3$ (Vertical velocity) | 50.0 |
| $b_4$ (Orientation) | 20.0 |
| $b_5$ (Pitch weight) | 1.0 |
| $b_6$ (Roll weight) | 1.0 |
| $b_7$ (Yaw weight) | 1.0 |
| $b_8$ (Vertical acceleration) | 5.0 |
| $b_9$ (Control effort) | 0.005 |
| Episode duration | 20.0s |
| Step time | 0.02s |

Figure 3 depicts the Gazebo environment during RL training, showing the biped robot navigating a planar terrain.

*1) Deep Deterministic Policy Gradient (DDPG) Algorithm:* In this study, the Deep Deterministic Policy Gradient (DDPG) algorithm—a model-free, offpolicy actor-critic technique intended for continuous action spaces [17]—is employed by the Reinforcement Learning (RL) controller. Because DDPG can analyze complex state as well as action spaces, it can optimize torques on joints or position inputs for stable locomotion, making it especially well-suited

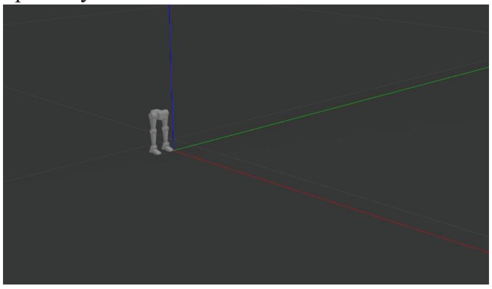

Fig. 3: RL Training Environment in Gazebo

for controlling bipedal gait. The DDPG algorithm, its elements, and the mathematical foundation for the gait control process are described in this section.

By combining the advantages of policy gradient techniques and Deep Q-Learning, DDPG overcomes the drawbacks of conventional RL algorithms in continuous domains. It uses two neural networks: a critic network that assesses the quality of those activities and a actor network that maps states to actions. The critic calculates the predicted cumulative reward (Q-value), whereas the actor chooses acts deterministically. The target networks and previous replay are used by DDPG to stabilize training, reducing problems like as sample correlation and overestimation bias.

The state space for the biped robot includes joint angles, joint velocities, Center of Mass (CoM) position, and foot contact states, represented as $s \in \mathbb{R}^n$. The action space consists of continuous joint torques or position commands, denoted as $a \in \mathbb{R}^m$. The reward function, defined in Equation (5), guides the learning process by balancing forward motion, stability, and energy efficiency.

• Actor-Critic Framework:

The deterministic policy $\mu(s|\theta^\mu) : S \rightarrow A$, which maps states to actions, is defined by the actor network, parameterized by $\theta^\mu$. The action-value function $Q(s,a|\theta^Q)$, that calculates the expected cumulative reward for doing action $a$ in state $s$ and then adhering to policy $\mu$, is approximated by the critic network, parameterized by $\theta^Q$.

Using the Bellman equation, the critic learns by minimize the temporal-difference (TD) error. A stateaction pair's target Q-value is calculated as follows:

$$y_i = r_i + \gamma Q(s_{i+1}, \mu(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) \qquad (5)$$

where $r_i$ is the immediate reward, $\gamma \in [0,1)$ is the discount factor, and $\theta^{\mu'}$ and $\theta^{Q'}$ are parameters of the target actor and critic networks, respectively. The critic loss is defined as:

$$L(\theta^Q) = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2 \qquad (6)$$

where $N$ is the batch size sampled from the experience replay buffer, which stores transitions $(s_i, a_i, r_i, s_{i+1})$.

The deterministic policy gradient, which maximizes the predicted Q-value, is used to update the actor. The following represents the actor's objective function gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s=s_i} \qquad (7)$$

This gradient updates the actor parameters to improve the policy by selecting actions that yield higher Q-values.

• Exploration and Stability:

During training, DDPG introduces noise to the actor's movements to guarantee exploration in the continuous action space. Temporally correlated noise is produced via an Ornstein-Uhlenbeck process, which is described as follows:

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t \qquad (8)$$

where $x_t$ is the noise state, $\theta$ controls the speed of mean reversion, $\mu$ is the mean, $\sigma$ is the volatility, and $dW_t$ is a Wiener process. The noisy action is computed as:

$$a = \mu(s|\theta^\mu) + \mathrm{N} \qquad (9)$$

where N is the noise sampled from the OrnsteinUhlenbeck process.

To enhance stability, DDPG employs soft target updates for the target networks, updating their parameters gradually:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}, \theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'} \qquad (10)$$

where $\tau \ll 1$ (e.g., $\tau = 0.001$) ensures slow tracking of the learned networks, reducing divergence. In this study, the DDPG algorithm is implemented using the StableBaselines3 library within the ROS framework, interfaced with Gazebo for simulation. The actor and critic networks are multilayer perceptrons with two hidden layers (256 and 128 units, ReLU activation). The learning rate is set to 0.001 for both networks, with a discount factor of $\gamma = 0.99$. The experience replay buffer stores 100,000 transitions, and a batch size of 64 is used for training. Domain randomization (e.g., varying terrain friction between 0.5 and 1.0) is applied to improve robustness. Training occurs over 1 million timesteps, with performance monitored via TensorBoard.

The reward function parameters (Table in Section V.B) are tuned to prioritize forward velocity while penalizing instability and excessive control effort. The DDPG algorithm's ability to optimize continuous actions enables the biped robot to achieve stable gait patterns and robust CoM stabilization on planar terrain.

## V. SIMULATION AND RESULTS

### *A. Cubic Spline with PID Control*

Simulations are conducted in Gazebo and MATLAB to evaluate both controllers. The cubic spline with PID approach is tested for a 1.3-second gait cycle, tracking hip, knee, and ankle angles with high accuracy (RMSE ¡ 0.05 rad). Figure 5 shows the tracking performance, with smooth transitions and minimal errors.

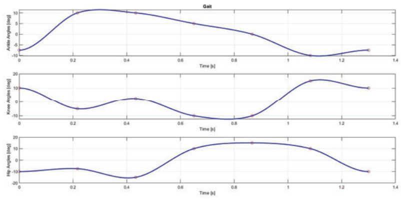

Fig. 4: Cubic Spline input trajectory

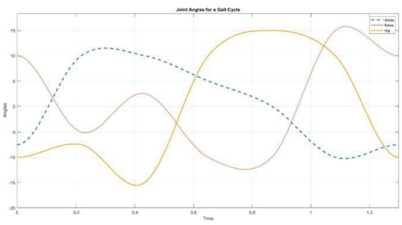

Fig. 5: Hip, Knee, and Ankle Angle Tracking

### *B. Reinforcement Learning Controller*

The RL controller is trained for 1 million timesteps, achieving stable locomotion with an average reward of 85 (Fig. 6). The reward increase indicates effective gait optimization, with the robot maintaining balance and consistent stride lengths (0.3 m) on planar terrain.

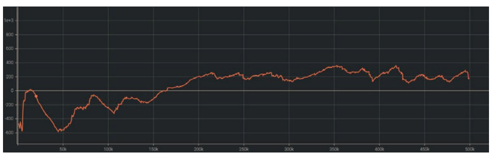

Fig. 6: Average Reward During RL Training

## VI. DISCUSSION

The cubic spline with PID control excels in precise joint tracking, making it suitable for controlled environments where predefined trajectories are sufficient. However, its reliance on fixed trajectories limits adaptability to dynamic or uneven terrains. The RL-based controller, while computationally intensive, demonstrates superior adaptability by learning optimal gait parameters through interaction with the environment. The DDPG algorithm's ability to handle continuous action spaces enables fine-tuned control, particularly for foot placement and CoM stabilization.

The trade-offs between the two approaches are evident: the PID-based method offers simplicity and reliability for structured tasks, while RL provides robustness in unstructured environments. The simulation results suggest that a hybrid approach, combining the precision of PID control with the adaptability of RL, could further enhance performance. For instance, using PID for low-level joint control and RL for high-level gait planning could leverage the strengths of both methods.

Challenges in RL training include long training times (approximately 10 hours on a GPU) and sensitivity to hyperparameter tuning. Future improvements could involve transfer learning to reduce training time or integrating sensor feedback (e.g., IMUs, force sensors) for real-time adaptation.

## VII. CONCLUSION

This study presents two gait control strategies for a 6-DoF biped robot: cubic spline interpolation with PID control for precise trajectory tracking, and reinforcement learning (RL) using the Deep Deterministic Policy Gradient (DDPG) algorithm for adaptive gait optimization. The PID-based approach ensures smooth and accurate joint movements with minimal tracking error, making it effective for structured environments. On the other hand, the RL-based approach demonstrates adaptability and robustness by learning optimal gait parameters that enhance stability in dynamic conditions. Simulations conducted in Gazebo validate the effectiveness of both control strategies, with the RL approach showing significant potential for operation in complex and unpredictable environments.

Building upon these findings, future work will focus on transitioning the controllers from simulation to realworld deployment by implementing them on a physical 6-DoF biped robot. This includes integrating inertial measurement units (IMUs) and force sensors to enable real-time feedback and improve interaction with the environment. Additionally, efforts will be made to enhance energy efficiency by optimizing the RL reward function to minimize power consumption during locomotion. Expanding the training environment to include more challenging terrains such as stairs, slopes, and uneven surfaces will further improve the robustness and versatility of the robot. These advancements aim to bring bipedal robots closer to practical applications in fields such as healthcare, rehabilitation, and disaster response.

## REFERENCES


[1] M. Acar et al., "Design of a 12-DOF biped robot," in *Proc. Int. Conf. Recent Measurements*, 2010, pp. 215–226.

[2] M. A. Adly et al., "Design and kinematic analysis of biped robot," in *2010 Int. Conf. Intell. Adv. Syst.*, IEEE, 2010, pp. 1–5.

[3] J. Ahn et al., "Versatile locomotion planning and control for humanoid robots," 2010, 2011, 2012.

[4] A. Babiarz et al., "Multimarginal model of a human leg: The switched linear system approach," in *2015 Int. Conf. Pervasive Embedded Com-*

[5] D. Bharadwaj and M. Prateek, "Kinematics and dynamics of lower body of autonomous humanoid biped robot," *Int. J. Innovative Technol. Exp.*, 2011, 2012.

[6] M.-A. Chadi and H. Mossannif, "Understanding reinforcement learning algorithms: The progress from basic Qlearning to proximal policy optimization," *arXiv preprint arXiv:2304.00026*, 2023.

[7] K. Chami, "Modeling and Designing of Bipedal Walking Robot," 2017.

[8] A. Chauhan and J. Kumar, "Analysis of Kinematics of a 12-DOF Biped Robot Gait by Parametrization of Its Body Trajectories," in *Recent Develop. Electron. Commun. Syst.*, IOS Press, 2023, pp. 104–112.

[9] J. Cho and J. H. Park, "Model predictive control of running biped robot," *Appl. Sci.*, vol. 12, no. 21, p. 11183, 2022.

[10] H. Daneshmand et al., “Variable horizon model predictive control for bipedal locomotion,” *Proc. IEEE Int. Conf. Robot. Autom.*, 2022.

[11] E. Dantec et al., “Whole-body MPC for humanoid locomotion,” *IEEE Trans. Robot.*, vol. 38, no. 4, 2022.

[12] A. Hereid et al., “Dynamic walking on stepping stones with gait library and control,” *Proc. IEEE Int. Conf. Robot. Autom.*, 2017.

[13] C. Hernandez-Santos et al., “Kinematics and dynamics of´ a new 16 DOF biped robot with toes,” *Proc. Int. Conf. Mechatron.*, 2017.

[14] W. Huang et al., “Quadrupedal principles for bipedal locomotion,” *Proc. IEEE Int. Conf. Robot. Autom.*, 2021.

[15] S. Jadlovska et al., “Model-based controller design for a´ compass gait biped robot,” *Proc. IEEE Int. Conf. Cybern.*, 2020.

[16] J. Jing and X. Zheng, “CoM trajectory control for stable bipedal walking,” *Proc. IEEE Int. Conf. Robot. Autom.*, 2020.

[17] V. Kumar et al., “Deep deterministic policy gradient for bipedal walking,” *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, 2021.

[18] Y. Leng et al., “Universal walking control framework for bipedal robots,” *Proc. IEEE Int. Conf. Robot. Autom.*, 2022.

[19] T. Mikolajczyk et al., “Advanced sensing in bipedal robots,” *Proc. IEEE Int. Conf. Sens.*, 2020.

[20] L. Miteva et al., “Anthropomorphic gait generation for bipedal robots,” *Proc. IEEE Int. Conf. Robot. Autom.*, 2021.

[21] I. Radosavovic et al., “Real-world humanoid locomotion with reinforcement learning,” *arXiv preprint arXiv:2304.00027*, 2023.

[22] M. Russo et al., “Kinematic modeling of humanoid torsos,” *Proc. IEEE Int. Conf. Robot. Autom.*, 2020.

[23] A. K. Singh et al., “Footstep planning with reinforcement learning for a biped robot,” *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, 2020.

[24] M. L. Strydom et al., “Kinematic model of the human leg using DH parameters,” *IEEE Access*, vol. 8, pp. 191737–191750, 2020.

[25] F. Sun, H. H. Ju, and P. Y. Cui, “A new 12 DOF biped robot’s mechanical design and kinematic analysis,” in *Proc. Off. Electron. Mech. Eng. Inf. Technol.*, vol. 5, IEEE, 2011, pp. 2396–2400.

[26] R. Szabo, “Design Approach for Evolutionary Techniques Using Genetic Algorithms: Application for a Biped Robot to Learn to Walk and Rise after a Fall,” *Mathematics*, vol. 11, no. 13, p. 2931, 2023.

[27] C. Vissumach, A. Sarhar, and P. Needharan, “Kinematic and dynamic simulation of biped robot locomotion on multi-terrain surfaces,” in *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 577, no. 1, p. 012094, 2019.

[28] Z. Xie, P. Gergondet, F. Kanehiro, et al., “Learning bipedal walking for humanoids with current feedback,” *IEEE Access*, vol. 11, pp. 82013–82023, 2023.

[29] K. Z. Z., “Teach biped robots to walk via gait principles and reinforcement learning with adversarial critics,” *arXiv preprint arXiv:1910.10194*, 2019.

[30] J. Zhang et al., “Structural design and kinematics simulation of hydraulic biped robot,” *Appl. Sci.*, vol. 10, no. 18, p. 6377, 2020.