Control and Gait generation of a 3D biped robot in Gazebo

Nahal Memar Kocheh Bagh

Master of Science by Research

University of York

Department of Computer Science

July 2024

1 Abstract

Today's bipedal robots still are behind humans in terms of efficiency, speed, and resilience in movement. Consequently, this thesis proposes a control strategy for dynamic walking inspired by human motion control principles. Central to this approach are the utilization of passive dynamics, hierarchical control mechanisms, and reflex actions, all achieved without necessitating a complete dynamic model. Walking stability is ensured through a series of postural reflexes linked to the extrapolated center of mass motion. It is demonstrated that only a minimal set of joints need simultaneous active control throughout different walking phases. Alongside detailing the control strategy, the thesis introduces an anthropomorphic biped model and highlights the importance of features such as compliant actuation for optimal walking performance. Specifically, the proposed method relies on joints with characteristics akin to the human muscle-tendon system, including non-self-locking, torque-controllable mechanisms with parallel elasticity and minimal friction. The effectiveness of the approach is validated through simulations of 3D dynamic walking, revealing an efficient, smooth, and rapid gait capable of handling significant disturbances. Moreover, the resulting joint trajectories closely resemble human walking patterns. In recent years, numerous researches have been done based on simulation of legged mechanism, especially on biped robots simulation and control. This research focuses on the biped robot simulation in Gazebo and the control of it over various trains such as flat environment, ascending and descending surfaces, passing a ditch and different obstacles with the aid of 3D modeling methods. the bipedal robot‘s movement is controlled by 3 different methods (Neural Network controller, Fuzzy Logic Control and PID controller). Finally the results of all these controllers are provided and the compression is done in later chapters.

Acknowledgments: First I would like to thank my Masters supervisor, Dr Pengcheng Liu for his continuous support through all my studies. He was always available to talk with me, provide feedback and discuss about new ideas. His involvement and skills played a crucial role in making this thesis what is. I also thank the referees for the time they spent reading this dissertation and accepting to review my work. I am glad that I have been able to exchange with my supervisor, whose work guided my scientific developments. My friends as a whole greatly contributed to making this masters by research an enjoyable experience despite the inevitable ups and downs that come with a long term scientific project. For all our discussions, activities and positive atmosphere, I thank them too. Finally, I would like to thank my parents for their support, not only for these years but as far as I remember. This defense is merely a step on a long road and they helped me pave it down.

Authors Declaration: I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for a degree or other qualification at this University or elsewhere. All sources are acknowledged as references.

Table of Contents

3 Chapter 1: the introduction

4 Introduction

Despite decades of research, bipedal robots still struggle to replicate the graceful motions and dexterity observed in human walking. Presently, most bipeds rely on analytical approaches rooted in multi body dynamics, pre-calculated joint trajectories, and Zero-Moment Point considerations to ensure stability. While these efforts have yielded impressive results in two-legged locomotion and other movement skills, they suffer from drawbacks such as strong model dependency, high energy and computational costs, and susceptibility to unknown disturbances. In contrast, human locomotion is characterized by elegance, robustness, speed, and energy efficiency. This dichotomy forms the basis for the two main hypotheses explored in this thesis: firstly, that a control system informed by insights into human motion control can endow bipedal robots with human-like walking capabilities, and secondly, that certain properties of human morphology are essential to fully exploit such a control system. To address these hypotheses, a control methodology is developed based on key features of human walking control and applied to a biped model incorporating human-like characteristics. While research on transferring control and morphology aspects from biology to walking machines has been ongoing, this approach aims to differentiate itself by the depth of biological analysis included, the manner in which these aspects are integrated into a robot control system, and the complexity of the target robotic platform under consideration. Bipedal robot control is one of the most challenging and popular research topics in the field of robotics. Despite significant progress, achieving the fluidity and adaptability of human walking remains a formidable challenge. Unlike classical control problems, such as industrial robot arm manipulation, the control of bipedal robots presents unique complexities. Various classical model-based control methods, including trajectory tracking, robust control, and model predictive control, have been proposed. However, these approaches often lack flexibility and struggle to ensure stability, adaptability, and robustness in bipedal robots. Several inherent characteristics of bipedal robots exacerbate these challenges: (1). Nonlinear Dynamics: Bipedal robots exhibit highly nonlinear and naturally unstable dynamics, rendering traditional linear control theories inadequate. (2). Dynamic Discontinuity: Walking cycles involve transitions between statically stable double-support phases and statically unstable single-support phases, necessitating adaptive control strategies. (3). Underactuation: Despite control over individual joints, bipedal robots remain underactuated due to their disconnectedness from the ground, posing additional control challenges. (4). Multivariable Systems: Bipedal locomotion involves numerous degrees of freedom, especially in three-dimensional spaces, complicating coordination and interaction between joints. (5). Changing Environments: Bipedal robots must adapt to uncertain and evolving environments, such as uneven terrain or obstacles, requiring rapid adjustments in

control strategies. Advancements in computing power have facilitated the implementation of sophisticated learning algorithms, offering promising avenues to address these challenges. Learning-based control, situated at the intersection of robotics and machine learning, enables robots to autonomously refine control policies through interaction with the environment. Unlike classical control methods, learning control techniques leverage partial models and iterative parameter tuning to acquire desired skills.

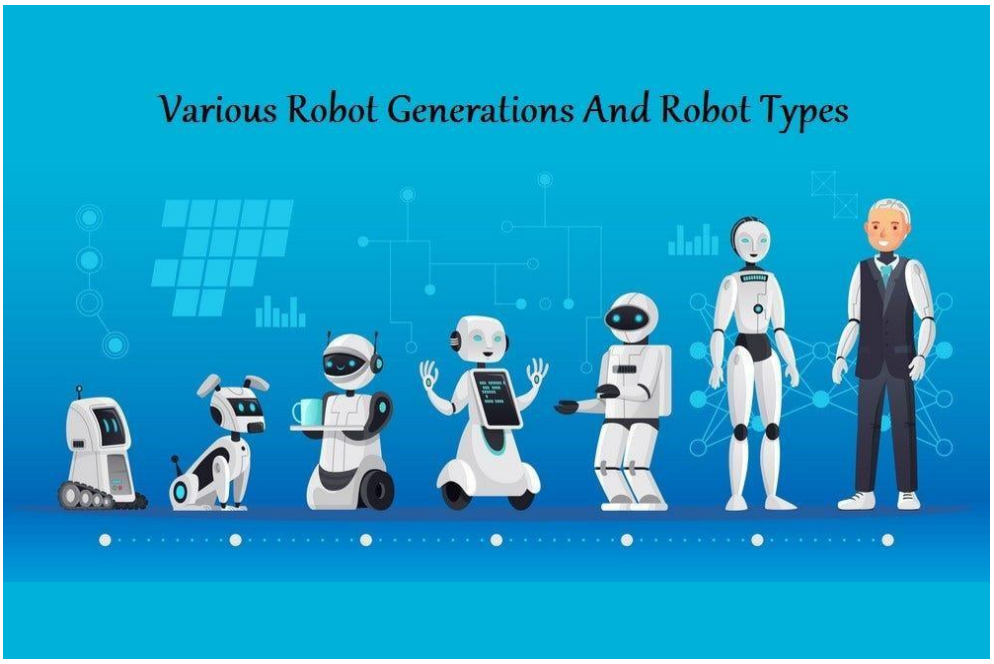

Fig. 1: Various robots and their link to a bipedrobot

5 Locomotion of a biped robot

Biped robots, characterized by their two legs, have the capability to walk on various surface and perform tasks similar to those carried out by humans. The locomotion of biped robots is significantly influenced by the gait cycle and the environmental structure in which they operate. The gait of a biped robot refers to the coordination between its legs and body movements during locomotion on specific surface. It can be classified into periodic and non-periodic gaits. Periodic gaits involve the repetitive generation of the same sequence of steps from start to finish. Non-periodic gaits, however, adapt their gait cycle based on environmental conditions. The walking cycle of a biped robot consists of two distinct phases: the single support phase (SSP) and the instantaneous double support phase (DSP). During the SSP, the robot takes a forward stride and covers a certain distance, while the double support phase is a momentary phase that allows for the exchange of leg support. Gait generation for biped robots can be achieved through two approaches: active walking and passive walking. Active walking involves attaching actuators to the joints of the robots' models, enabling controlled movements. In contrast, passive walking does not involve actuators and relies on natural dynamics and momentum [1]. Two main types of bipedal walking systems are static walking and dynamic walking. In static walking, the balance of the biped robot is determined based on the center of mass (COM). On the other hand, dynamic walking involves a faster walking cycle compared to static walking, with the balance of the biped robot assessed based on the zero-moment point (ZMP). The zero-moment point represents the point around which the sum of all moments generated by active forces equals zero. The balance of a biped robot can be measured using the concept of dynamic balance margin (DBM), which evaluates the robot's ability to maintain stability during walking. The introduction of the concept of Zero Moment Point (ZMP) by researchers Vukobratovic and Stepanenko [2] has played a significant role in gait generation for biped robots. They considered the upper body of the biped walking model as an inverted pendulum, with the Zero Moment Point aid- ing in the determination of the Dynamic Balance Margin (DBM). The Dynamic Balance Margin provides an estimation of the robots stability in dynamically balanced systems. Various techniques have been employed to compensate for the ZMP and ensure stability, including preview control, AI-based gait generation, and model predictive control. Additionally, researchers have explored periodicity-based gait, capture point theory, and foot placement estimators to analyze dynamic stability in biped robots [3].

For ZMP (Zero Moment Point):

Sagittal plane: $x_{\text{ZMP}} = \frac{\sum_{i=1}^{n}(I_i\theta_i + m_i\ddot{x}_i(\tilde{z}_i - g) - m_i\ddot{z}_i)}{\sum_{i=1}^{n} m_i(\tilde{z}_i - g)}$ (1)

Frontal plane: $y_{\text{ZMP}} = \frac{\sum_{i=1}^{n}(I_i\theta_i + m_i\ddot{y}_i(\tilde{z}_i - g) - m_i\ddot{z}_i)}{\sum_{i=1}^{n} m_i(\tilde{z}_i - g)}$ (2)

For DMB (Dynamic Margin of Balance):

Sagittal plane: $x_{\text{DBM}} = \frac{l}{2} - |x_{\text{ZMP}}|$ (3)

Frontal plane: $y_{\text{DBM}} = \frac{w}{2} - |y_{\text{ZMP}}|$ (4)

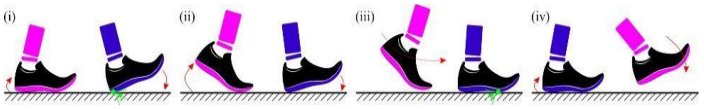

Fig. 2: Gait phases (i) SSP ends, DSP begins, (ii) DSP, (iii) DSP ends, SSP begins (iv) SSP[4]

6 MOTION GENERATION FOR DYNAMIC BIPEDAL LOCOMOTION

Throughout the previous section, we outlined how the locomotion problem is fundamentally different from traditional approaches to modeling fixed-base robots. It is because of this inherent complexity that virtually all approaches to realizing dynamic walking must transcribe the locomotion problem into a motion planner that can handle the various constraints naturally imposed on the problem. While several of the more classical walking paradigms offer simple solutions to conservative walking, there has been a push over the last two decades toward leveraging optimization to obtain increasingly dynamic maneuvers [5].

6.1 Step Planning with Linear and Reduced-Order Models

For the simplest models of walking, such as traditional ZMP and LIPM versions of the capture point, the linear dynamics of the restricted system of- ten yield straightforward approaches to planning the motion of the COM. The walking characterized by these linear models often implicitly satisfies quasi- static stability assumptions, ultimately allowing a control designer to decouple the high-level step planner and low-level balance controllers [6]. vein, Kajita et al. [7] introduced the jerk of the COM as an input controlled by a discrete linear quadratic regulator controller with preview action to plan ZMP trajecto- ries for predefined footsteps. However, predefining the motions of the ZMP or

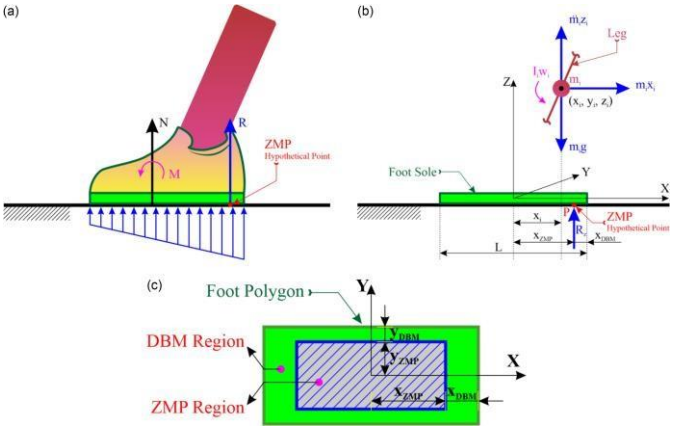

Fig. 3: (a) Schematic diagram showing the ZMP acting on the foot support, (b) free body diagram showing all forces responsible for creating moment about ZMP, (c) schematic diagram showing the range of possible ZMP region and DBM region under the foot polygon [8].

footholds is not always necessary or desirable. If planners for these simple models could instead be performed online, then the robot may be able to mitigate issues related to reactivity. Wieber [9] proposed using linear, trajectory-free model predictive control as a method for explicitly handling the constraints imposed by the ZMP approach while continuously reevaluating the walking path. Stephens and Atkeson [10] presented the use of model predictive control for push recovery and stepping on the SARCOS humanoid, which could be extended to obtain walking behaviors. Studies have also shown how optimization and model predictive control can extend the notions of capture point to viable regions on which the biped can step and how push recovery can be planned over a horizon of multiple steps [11]. Despite the ability of these planners to adapt online, they cannot handle the discrete dynamics associated with foot strike, and they demand near-zero impact forces, which rules out the nontrivial impacts that are naturally associated with dynamic walking. It is also difficult to provide a priori guarantees on whether any given reduceorder plan is feasible to execute on the full-order dynamics [12]. Such methods typically use inverse kinematics or inverse dynamics, sometimes in an operational-space formulation, to compute the full-order control inputs at each instant [13]. Solving such near-term inverse problems does not imply that future inverse problems in the trajectory will be feasible, which requires additional planning.

6.2 Nonlinear Optimization for Gait Generation

As a result of the rapid developments within the trajectory optimization community, researchers began to move toward utilizing nonlinear dynamic gait optimizations rather than relying on the constraints imposed by linear modeling assumptions. The use of nonlinear optimization (i.e., numerical approaches) to generate stable walking behaviors on bipeds is not a new concept [14], though computational limitations were a considerable hindrance to generating motions on 3D robots. During the mid-2000s, computation power finally in- creased sufficiently to begin handling 3D dynamic walking behaviors [15].

6.3 Open-loop optimization

The use of open-loop optimization to generate feasible motions for actuated robots is a natural extension of approaches used throughout the field of trajectory optimization, where the planning problem is seen as decoupled from the feedback control applied to the actual robot [16], and approximately optimal solutions are often sufficient. Furthermore, in recent years, the application of advanced trajectory optimization methods such as direct collocation has made the optimization of the full-body dynamics of Equation 1 more computationally tractable, sparking a growing interest in considering the full-body dynamics of the robot in the planning problem. For instance, in order to control the open-loop trajectory that results from the direct collocation optimization, a classical linear quadratic regulator–based feedback controller can be constructed to stabilize the resulting trajectory obtained for the constrained dynamical system. Complementary Lagrangian systems formed the basis of the approach used by Posa et al. [17], which allowed the optimizer to find walking behaviors without a prior enumeration of the type and order of contact events. Open-loop trajectory optimization has also been used to satisfy ZMP conditions in a nonlinear fashion, which considerably improved the dynamical nature of the conservative walking.

6.3 Closed-loop optimization

While the preceding nonlinear optimization approaches do consider the full-body dynamics of the robot, it is not always desirable to apply feedback controllers to stabilize an approximately optimal open-loop plan. Rather, it is often beneficial to couple the gait generation and controller synthesis problems into a single framework: closed-loop optimization. This framework allows, among other things, the generation of provably stable walking behaviors that simultaneously satisfy the constraints on the system from admissible configurations to torque bounds. This idea forms the basis of designing walking gaits with the HZD method, where feedback control is used to generate provable stable periodic orbits [18].

7 Gait generation and design issues of the biped robot

This part covers the proposed taxonomy for various aspects of gait generation and the design of biped robots. we have attempted to simplify and classify the concerning factors and gait generation techniques for the gait generation prob- lem and design issues of the biped robot based on various perspectives through the proposed systematic taxonomy. This can help to identify the problem and then wisely select the appropriate strategy [19].

7.0.1 Gait GENERATION TECHNIQUES

Gait Generation Techniques

1. **Model-Based Gait**
   - **Interpolation/Reference Trajectory-Based Gait**: Relies on predefined trajectories to generate motion.
   - **Inverted Pendulum Model (IPM)**: Utilizes the dynamics of an inverted pendulum, including linearized IPM approaches, for optimizing gait parameters.
2. **Natural Dynamics-Based Gait**
   - Employs biped models enhanced with virtual components such as springs and dampers to replicate natural walking dynamics.
3. **Biological Mechanism-Based Gait**
   - **Human Motion Capture Data (HMCD)**: Mimics real human movements for naturalistic gait generation.
   - **Central Pattern Generators (CPG)**: Neural network-based models that replicate rhythmic and periodic motion.
   - **Biologically-Inspired Algorithms**: Leverages mechanisms inspired by natural motion principles.
4. **Stability Criterion-Based Gait**
   - **Zero Moment Point (ZMP) Criterion**: Ensures balance by keeping the center of pressure within the support area.
   - **Center of Pressure (CoP) Criterion**: Focuses on foot-ground interaction to maintain balance.
   - **Center of Gravity (CoG) Criterion**: Tracks the center of mass for stability.
   - **Foot Rotation Indicator (FRI)**: Monitors foot placement to assess stability.
   - **Periodicity-Based Gait**: Includes methods such as limit cycle analysis to maintain consistent rhythmic motion.

Factors Influencing Gait

1. Trajectory Planning

- Based on complex mathematical equations for path and motion generation.

2. Terrain Adaptability

- **Structured Terrains**: Motion planning for predefined surfaces, such as:
  - Flat ground
  - Ascending and descending slopes
  - Staircases
- **Semi-Structured Terrains**: Tackling minor irregularities, such as:
  - Obstacles on flat ground
  - Small ditches
- **Unstructured Terrains**: Adapts to unpredictable or uneven environments.

3. Trajectory Types

- **Polynomial Trajectories**: Includes quadratic (second-order), cubic (third-order), quintic (fifth-order), etc.
- **Cyclical Trajectories**: Repeating patterns for consistent motion.
- **Bezier Trajectories**: Smooth path generation for complex movements.

4. Path Planning

- Avoidance of static obstacles.
- Navigation around dynamic obstacles for smooth and adaptive motion.

8 Chapter 2: Gait generation Methods

Fig. 4: Taxonomy for various design issues of the biped robot

9 Types of gait generation techniques

Fundamentally, there are four gait generation techniques; model-based, natural dynamics-based, bionic kinematics or biological mechanism-based, and stability criterion-based technique The model-based gait generation technique mainly consists of interpolation-based gaits which means generating reference trajectories by polynomial satisfaction of the constraints and tracking them by using the control system [20]; linear inverted pendulum model (LIPM) dynamics- based modeling and optimization-based gaits that include the optimization of energy consumption, robot construction, control system, and adaptation. The drawback of this technique is the requirement of all the information on the dynamic parameters of the respective biped model [21].

It is to be noticed that the biological mechanism-based gait generation is inspired by animal and human motion capture data (HMCD) which can generate different stable rhythmic patterns along with the capabilities to change the pattern and its speed quickly [23]. Further, the central pattern generators (CPG) and neural networks (NN) are inside the spinal cord, capable of generating rhythmic locomotion and lacking any sensory signals. The Matsuoka neural oscillator [24] and the Van der Pol oscillator [25], two popular models, are used for modeling the CPGs. It has been observed that the other approaches which are biologically inspired fall under artificial intelligence (AI)-based gait, which encompasses genetic algorithms (GA), fuzzy logic (FL), and NN.

Moreover, natural dynamics-based gait is performed based on intuitive control, natural dynamics of the biped, physics of the system, and virtual elements like dampers and springs that is why this technique does not need any predefined reference trajectories [26]. And also, the gait can be performed based on stability criterion-based gait generation techniques, including ZMP, DBM, CoP, COG, CoM, FRI, theory of capture points, foot placement estimator, periodicity-based gaits, and limit cycle analysis.

Apart from these gait generation techniques, the researchers should also identify some factors while planning the gait generation of the biped robot, such as a suitable trajectory equation based on polynomial, cycloidal, and Bezier curves, avoiding static and dynamic obstacles for path planning, and types of terrain for estimation of the boundary conditions for swing foot, hip, and wrist trajectories.

9.0.1 Biped robot design and its challenges

The above discussion of fundamental gait generation techniques may help the researchers to identify standards and best practices for developing the biped robot and generating its gait on different terrains. The fundamentals of biped locomotion have been explained briefly in the introduction part of this thesis. At the same time, the design issues of the biped robot have been presented previously, which needs to keep in mind before planning and designing the biped robot. These factors also affect the ability of a robot to walk over uneven terrain. The fundamentals of modeling the biped robot, such as deciding the number of degrees of freedom of the robot, include the allocation of the actuators and their orientations [27]. Further, the type of trajectory must be planned for each part of the robot's mechanism, such as the swing foot, wrist end, and hip, to enable the robot to move from the source to the aimed position, that is, path planning. The analytical modeling could be of the planner type, which includes trajectory planning only for the sagittal plane, whereas trajectory planning for both the sagittal and frontal planes or the sagittal, frontal, and horizontal planes, that is, 3D modeling must be done to enable the robot to walk in a real environment. In addition to, researchers discussed various types of walking patterns, type of

foot and ground contact, including probable forces developed due to the impact of the heel on the ground, and arrangement of heel and toe contact with the ground when planning the dynamics for improved stability robustness. Other than these, elastic and stiff links of robot may benefit with some flexibility to absorb impact and cause instability because of the uncertain motions generated by the elasticity factor [28]. However, the mobility of the links depends on active or passive joints. Moreover, various stability criteria as per their skills, such as ZMP, CoP, COG, FRI, can be possible to adopt. Therefore, the mathematical model of the biped robot consists of kinematics and dynamics by using any high-level programming language and also build their planned model with the help of software, such as CoppeliaSim, ROS, MATLAB, to do simulation and verify the feasibility of their planned model. Additionally, the researchers must concentrate on the characteristics of environments or terrains, optimization algorithms, autonomous navigation through biologically inspired learning algorithms for suitable decision-making and adaptively, designing the controller as per the nonlinearity present in the robot's mechanism, suitable gait generation techniques, the number of under- actuated and over actuated joints, planning the robustness against the probable unbalanced external forces, and both online and offline modes of tracking the deviation of the trajectories from the planned mathematical model [29]. In addition to planning includes the hardware of the robot consists of structure of the robot, sensors for recording real-time data, and microcontrollers for operating the actuators, which could be electric, hydraulic, or pneumatic drives. Over and above, some challenges observed while designing the biped robots are as follows:

• The biped robot joints are underactuated during SSP and overactuated during DSP [30]. Consequently, the dynamics and control laws are also changed during this phase transition. In addition to the above problem, the biped robot acts as an open chain mechanism in SSP and a closed chain mechanism in DSP, which consequently changes the dynamics equations to be used.

Over-actuation of the biped mechanism is controlled by kinematic Jacobian [31] and minimization of the joint torques by algebraic optimization [32]. However, the continuous dynamic response can still not be guaranteed [33]. Similarly, the problems that occurred during the SSP phase are encountered by using the four control techniques: port-Hamiltonian method [34], differentially flatness-based approach [35], hybrid zero dynamics (HZD) [36], and time scaling method [37].

• A humanoid robot can be made up of more than 30-DOF, which makes its stability and control more complex [38].

• When the biped robot is walking in various unknown environments in real time, it needs to develop robust algorithms for possible external disturbances and noises.

• Unexpected shocks and instability of the robot are happened due to stiff joints [39]. Many researchers have used elastic joints to overcome the problem, which can be preferred and consequently increase the system's DOF due to flexible joints. Remark: Despite the complexity of making the biped mechanism closer to imitating human walking, compliant legs are employed [40].

• The foot-ground contact needs to be designed appropriately to avoid impulsive forces. To overcome these challenges, researchers suggest to use a suitable controller other than the right selection of gait generation methodology.

10 Controllers used for generating the smooth gait

Many researchers have developed various control algorithms to control the motions and dynamic balancing of the biped robots for smoothly coordinated motions among the different mounted motors in every joint. researchers have discussed popular controlling techniques such as PID, CTC, NN, CMAC-NN, FLC, MPC impedance control in this section [41].

10.1 PID controller

The proportional integral derivative (PID) controller is most famous for industrial applications. The PID controller consists of proportional, integral, and

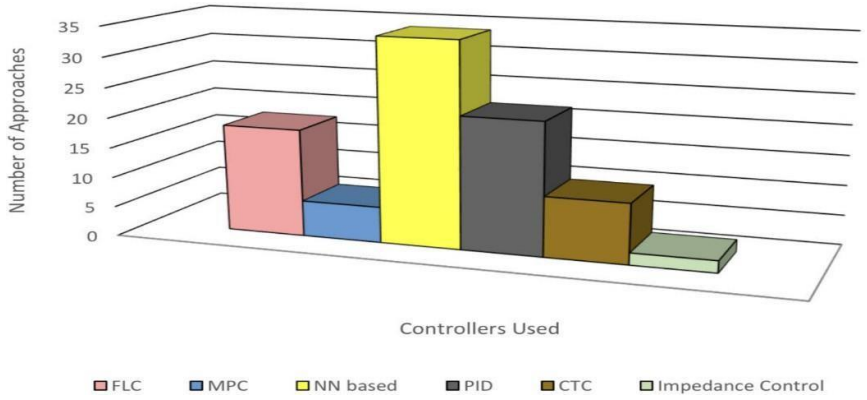

Fig. 5: Various controllers applied for biped gait generation on various terrains

derivative gains containing errors between target and achieved values. The PID controllers have been used widely for many years in the robotic field due to their simplicity and ease of controlling the controller's proportional, integral, and derivative gains. This PID controller can be simplified as PI (Proportional Integral), PD (Proportional Differential), and ID (Integral Differential) controllers for minimizing the nonlinearities in the system. The tracking system based on PD (Proportional Derivative) controllers allows adaptivity to the system as per the parameter variation and external forces [42]. Above 80 percent uncertainty level, the PID controller fails to provide a stable gait generation of walking on flat terrain and staircase for single support and biped-in-air phases [43]. the cascaded control is discussed with all the versions of P, PID-PI, SISO, MIMIO MIMO-SISO cascaded controllers in detail. The constant gain of PID controller consequences to very high speed then slowly reduced while adaptive gain of PID controller consequently smooth operation of a biped robot [44]. The tracking error convergence rate is controlled by the PD controller for the continuous task of the swing leg and subdue the nonlinear impacts by HZD assumptions for the discrete assignment of foot impact on the ground [45]. Several well-known optimization techniques have been used to optimize the gains of the PID controller like NN, FL, GA, PSO, DE and MCIWO, with NN being the most used one.

10.1.1 Computed torque controller (CTC)

Other than this, the CTC is an efficient way to generate dynamically stable gaits that curbs the system's nonlinearities [Reference Song, Yi, Zhao and Li66]. It can stabilize but requires an exact dynamic model of the robot mechanism. That factor puts limits on its applications. It can be described as a position-

oriented control technique. It is also called an inverse dynamics controller; it is one of the most widely used controllers. It was first introduced by a NASA scientist B. Markiewicz [46] in 1973. It is based on the principle of feedback linearization, a technique for simplifying a nonlinear model into a linear one. All nonlinearities and cross-coupling terms are calculated and eliminated in this method [47]. Its ability to transform a coupled, nonlinear mechanical system into a linear, decoupled, and stable system is one of its appealing qualities. As a result, the researchers are able to control nonlinear systems using linear controllers like PD and PID controllers. Accurate dynamical models of robotic manipulators are necessary for the CTC scheme [48] which puts limitation on its usage. Song et al. [49] made an effort to address this issue and proposed a method for trajectory tracking issues of robotic manipulators with structured uncertainty and/or unstructured uncertainty by integrating CTC and Fuzzy Control. Since the parameters of the majority of physical systems are either unknown or time-variant in reality, a computed torque-like controller is used to correct the dynamic equation of the robot.

10.2 NN controller

The NN technique ensures closed-loop execution for controlling the bounded errors. The NN have offline and online real-time learning characteristics for easy implementation [49]. The NN-based controller has been integrated with the cerebellar model articulation controller (CMAC) in most of the approaches, which is an integrative memory-type NN that was initially introduced by Albus [50]. Since then, it has been used in robotic applications for reinforcement learning architectures. It is a kind of NN which employs associative memory. It simplifies the large size of NN and its inherited problems. The CMAC executes better than the usual NN in terms of learning speed and is simple in computation and easy to implement. The NN involves entirely connected neurons, and all weights need to be updated in each learning cycle, which makes the NN slow. In contrast, CMAC is based on associative memory networking, but NN is more universal than CMAC [51].

10.3 Fuzzy logic controller (FLC)

The FLC system is a control scheme that investigates the input parameters. It considers them as logical data from 0 to 1, representing false and actual values, respectively. Still, the FL does not represent exact true or false but partially accurate values since it varies from 0 to 1. It was first introduced by Lotfi A. Zadeh [52]. The FLC is heuristic in nature, consisting of a knowledge base and human thinking for reducing nonlinearities. Heuristic characteristics cannot be implemented with traditional techniques. The FLC does not require accurate mathematical modeling and perfectly designed inputs to reduce the nonlinearities better than most controllers.

10.4 Impedance controller

The impedance controller is a dynamic control approach based on controlling the force and positions of the links. Controlling the impedance of any mechanism is controlling the force offered by the surroundings against the motions. It is being used in robotics, where the force and position of every link are essential in maintaining the dynamic stability and robustness to perform any gait. It was first introduced by Hogan [53] in 1984. By incorporating a feedback control algorithm for imposing a desired Cartesian impedance on the end effector of a nonlinear manipulator. The proposed method for controlling the dynamic behavior of a manipulator with its surroundings. With the help of this algorithm, it is no longer necessary to solve the inverse kinematics problem to control the robot's motion. Further, its unique characteristics allow the researcher to superimpose different controller actions for performing diversely targeted tasks. In addition to, the structure's ability while resist motion un- der any harmonic force is known as its mechanical impedance which is ratio of applied force (i.e., potential) to resulting motion, that is, (flow) [54]. The magnitude of force required to achieve a given velocity decreases as the swing's admittance decreases. The main goal of impedance controller is to control both the robot's motion and its contact forces.

10.5 Model predictive control

This technique is a broad control strategy that satisfies the system's constraints and gives optimum responses. In this technique, the reference trajectories are provided, based on which it predicts the future progression of the model. It itself is a broad research topic, and it has often been used in robotics. Lee and Markus discussed the significance of model predictive control [55]. Later, Shell Oil engineers developed the model predictive control technique in the 1970s and applied in 1973 [56]. Despite the significant computational load, the MPC outperforms structured PID controllers in terms of changes in system parameters (robust control), and very easily it can be applied to complicated multi variable. On the basis of immediate state evaluations and anticipated process responses, it can calculate the best possible control actions [57]. Due to these characteristics, it is suitable for sophisticated multivariate process control systems. The architecture of MPC depends on an integrated linear or nonlinear model for capturing the dynamic behavior of the process and predicting its response over a finite horizon window in order to assess the best control trajectory by resolving a dynamic optimization problem while taking input and state constraints into account at each sampling time.

11 Gait generation on various terrains

Balancing the two-legged robot is more complicated than the wheeled robot. Moreover, the locomotion of the legged robot has more universal appeal than wheeled locomotion due to its complex and remote applications based on different terrains where wheeled mobility is impossible. In this part we reviewed many approaches while generating the gait on different terrains such as flat, slope and staircase.

Table 1: an over view to all the gait generation approaches on different terrains

Terrain Type	Key Challenges	Gait Generation Methods	Notable Techniques
Flat Terrain	Balancing, trajectory synthesis, foot-ground interaction	Model-based (e.g., Linear Inverted Pendulum Model), Natural Dynamics, Stability Criterion, Bionic/Biological Mechanism	Polynomial interpolation (cubic, fourth-order), ZMP control, virtual height inverted pendulum, NN optimization for swing leg trajectory.
Slopes	Maintaining stability with varying inclinations	Stability Criterion (e.g., ZMP-based), Biological Mechanism-based (e.g., CPG-inspired), AI algorithms for adaptability	ZMP trajectory adjustment, hybrid approaches integrating neural networks with optimization methods like PSO.
Staircase	Adjusting for step height and swing phase coordination	Sliding Mode Control, NN-based controllers, and Stability Criterion	Enhanced DBM with cubic trajectory optimization, adaptive PID control for trajectory adherence.
Obstacle Navigation	Perception-based control, avoiding dynamic/static obstacles	Path planning and perception-based techniques using AI	NN trained by GA/DE, hybrid controllers integrating sensory data with optimization (e.g., regression controllers).
Ditches	Crossing distances wider than the leg's length	Analytical Modeling, NN and Fuzzy Logic Optimized by Genetic Algorithms	Adaptive gait planners for dynamic balance margin optimization, trajectory-based methods for efficient energy use.
Uneven Terrain	Uncertainties in surface patterns, maintaining dynamic stability	LIPM-based Modeling, Adaptive Algorithms, Hybrid Controllers	Dynamic balancing via real-time adjustments (e.g., impedance controllers), hybrid intelligence (fuzzy-NN) for adaptive navigation.
Unknown Environments	Real-time decision-making based on incomplete data from sensors	Reinforcement Learning (RL), Neural Networks, Environment Mapping and SLAM	RL-trained CPG networks, vision-based navigation (RGB-D sensors), adaptive controllers for terrain mapping and locomotion optimization.

11.1 Gait generation on the flat terrain

While performing the gait of the biped robot on a flat surface, several issues need to be fulfilled to complete one walking cycle. The most crucial aspects of walking are balancing, controlling, trajectory synthesis, and foot-ground interaction. Figure. 6 is the gait generation of the biped robot on a flat surface with interpolation of cubic polynomial trajectory for the swing leg [58]. Various

methodologies for biped gait generation are being discussed here based on four fundamental gait generation techniques adopted by researchers.

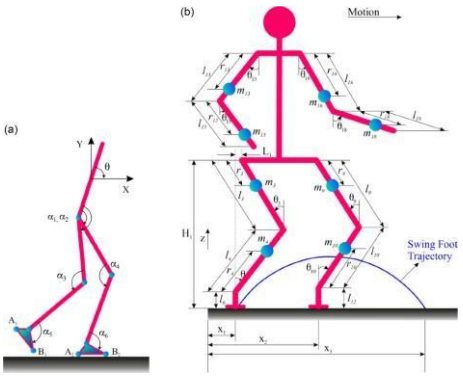

Fig. 6: Schematic diagram showing mass, length, and angles of each links of a biped robot walking on the flat terrain

Most of the researchers adopted the model-based gait technique for walking on flat terrain, which is the simplest case compared to any other terrain. The dynamics laws for the biped robot were determined by using the fundamentals of LIPM [59], virtual height inverted pendulum mode [60], Euler- Lagrange formulation [61], Newton-Euler approach, and then after calculation of the dynamics, the whole-body gait can be generated by using forward and inverse kinematics [62]. The complexity of the biped modeling can be dealt by arranging the hip, knee, and ankle joints of the biped model as under actuated and frictionless [63]. Interpolation of the joint trajectory is also adopted for gait generation. Chevallereau et al. [64] obtained optimal joint reference trajectories gait cycle by using fourth-order polynomial functions for joint variables while keeping ankle joint under actuated. The reduced ankle power was compensated by the motion of swinging leg and body for proper foot contact with the ground, smooth walking was obtained for the lesser complex biped model, and also dynamically stable bipedal gait over the flat terrain was obtained. Similarly, scholars [65] assigned cubic polynomial trajectories for the swing foot, hip, and wrist joint of 18-DOF humanoid robot.Numerous researchers have developed many control schemes to reduce the effect of nonlinearities in the biped mechanism due to complex dynamics. There- fore, nonlinear feedback control for 5-DOF biped model while moving in air and free fall motion; closed-loop Eigen structure assignment for prescribed gait of 5-DOF model [66]; two-level control scheme for generating prescribed gaits and motion to reduce large deviations; control scheme based on a novel integration of the multiple input multiple outputs (MIMO) framework for 10-DOF biped model. control technique by selecting state variables dependent output functions such as angular orientations and velocities along with Pfaff-Darbous principle and differential geometric tools [67]; robust control technique based on series elastic actuation in “FLAME” “TUlip” for limit cycle walking; local feedback at each joint of the robot [68] and feedback control scheme for stable cyclic gait developed to obtain the dynamic stability of the biped walking on flat terrain. Other than these, a technique based on wireless monitoring and controlling of actuators and sensors by employing the tunneling method was introduced by Nicolau et al. [69] for robot YABIRO, which is done by employing the tunneling method for enclosing CAN messages into a TCP/IP network over WiFi. Similarly, they developed an online adaptation technique based on a set of intuitions for tracking reference trajectories [70]. Researchers have also attempted to optimize the energy consumption for obtaining the periodic gaits by using Hamilton-Jacobi-Bellman type equations and obtaining the gait. It has been found that the gait transition from running to regular walking by releasing extra energy while shortening the legs with the help of an antagonistically driven hip joint consisting of two nonlinear springs, two AC servo motors and one free joint. Furthermore, Ji et al. [71] investigated the impulsive effects of the ankle push-off byaccelerating the swing leg and decreasing the changes in COM speed

to increase the gait speed. When the model's physics helps to generate the gait, it is termed natural dynamics-based gait. In the initial time of biped development, the researchers preferred the physics-based gait due to the unavailability of intelligent techniques such as passive pendular gaits in the swinging phase; forward and reverse walking of BIPER-1, 2, 3, 4 5. virtual spring and a damper to the prevalent inverted pendulum-based biped robot; intuitive gait strategy for 9-DOF biped model controlled by forces and posture and the virtual constraints for the able gait of 7-DOF biped model RABBIT [72]. A few approaches related to stability criterion-based gait were also reported. For mimicking human motions by tracking ZMP trajectories and generating stable gait by modifying the horizontal COG positions efficiently with the help of dispersed force sensors, motion capturing mechanisms, and force plate. Further, the they obtained the ZMP by using the universal force-moment sensor on WL-12RIII. In addition, Tagawa and Yamashit [73] introduced the Zero Moment Joint (ZMJ) concept and showed a stable biped gait for the 8-link biped model when ZMJ was the only ankle joint. Apart from the above-discussed approaches, few researchers have also generated the biped gait inspired by bionic or biological mechanism-based gaits, which are discussed here. Such as, Yazdani et al. [74] developed a bi-layer controller consisting of high-level and low-level controllers. The high-level controller utilizes all sensory information to deal with the dynamics and produce stable rhythmic motions through conscious learning during training. The low-level controller consists of a control network in which every individual node is an oscillatory dynamic that learns and reproduces the desired paths. The critic agent in the node allocates a particular controller for any parameter based on its eligibility. The proposed controller proved robust and stable as a dynamic controller but mainly featured as a path or trajectory-based controller. Similarly, the nonlinear oscillator has also been used to observe the sensor output to obtain real-time online trajectories.

11.2 Gait generation on ascending and descending the sloping terrain

Gait generation of the biped robot on a sloping surface is a more challenging task than the flat terrain. Figure.7. show the biped robot's gait generation in ascending and descending the sloping surface. Very few researchers have reported the model-based gait generation approach for ascending and descending sloping terrain. Kuo [75] developed an analogy of human gait with an inverted pendulum, provided a circular trajectory instead of a horizontal trajectory for COM, and found that a horizontal COM trajectory consumes more muscular energy. In addition to what has been said, Pratt [76] presented the natural dynamics and inherent robustness of the biped locomotion mechanism and developed the Spring Flamingo robot using a low-impedance controller which can start and stop while moving on slopes and rolling surfaces with various speeds. The derived control algorithm exhibits three stages: the primary algorithm control walking, the secondary algorithm exploits the kneecap, ankle, and passive swing leg natural dynamics, and the tertiary algorithm ensures fast walking of the swing leg. Scholars added lateral balance to the three- dimensional algorithm and simulated the 3D model.

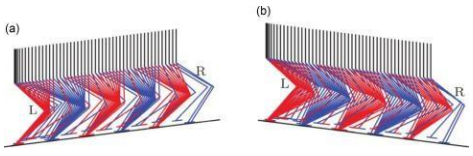

Fig. 7: Gait generation on a sloping surface (a) ascending the slope and (b) descending the slope

Stability criterion-based approaches have also been developed and imple-

mented for gait generation on a slope. Massah et al. [77] used 3D inverted pendulum-based equations and ZMP concept for developing a trajectory planner by employing the semi-ellipse EOM (equations of motion) for an NAO humanoid robot and simulated on Webots while walking on various slope terrains. Vundavilli and Pratihar [78] used ZMP concept and reported more DBM for ascending the slope than descending the slope. Furthermore, Hwang et al. [79] obtained momentum equations based on ZMP by treating biped robot as a pthesis and assuming the motion of CoM parallel to the slope and then simulated it by using ResurDyn and MATLAB commercial software. In addition to, Ito et al. [80] reduced the number of actuators of biped robot without sacrific- ing adaptability and ability then applied the gravity compensation mechanism and feedback from CoP of the ground reaction forces. Most of the researchers have attempted the biological mechanism-based gait for ascending and descend- ing the sloping terrain. The central pattern generator (CPG) has inspired the researchers to build learning architecture for biped robots of different configurations and enabled the biped robots for autonomous biped gait; smooth gait transition from flat to slope vice versa; walking on a flat plane with different friction properties and little change in inclination, stable gait on unknown inclination, and adaptively in different environments. Further, few gait generation algorithms have been developed for generating complex gait patterns using AI techniques such as Genetic Algorithm, neurons and neural pathways, genetic-neural (GA-NN) and genetic-fuzzy (GA-FLC), NN integrated with modified chaotic invasive weed optimization (MCIWO), and PSO algorithm. The AI has enabled the biped robots to walk on sloping terrains more efficiently, but if there is a change or increase in inclination angle, then some essential sensors must be attached to the biped legs. The researchers have employed the integration of position sensors (on joints) and force sensors (under foot) to identify slope gradient; gyroscope and accelerometer sensors to identify the upper body's posture; inertial measurement unit (IMU) sensor; and 2-axes accelerometer sensor for obtaining the smooth, balanced gait on slope terrain. To overcome the difficulty due to complex mathematical modeling, they developed a collective balancing reflex of threshold, PID, and hybrid control with a 2-axes accelerometer sensor, which does not need any mathematical modeling[81].

11.3 Gait generation on ascending and descending the staircase

The gait generation on the staircase is very different from the flat and slop- ing surfaces due to the approximate relationship between the height-width of every step and the length of the robot's leg. There are chances of collision of the robot with the staircase. Therefore, the swing phase take-off mechanism is essential in determining the gait pattern characteristics [82]. The synchronization of all robot links and defining the proper foot trajectory become vital for stabilizing the robot. Figure.8 show the gait generation of the biped robot for ascending and descending the staircase, respectively, by controlling the forward
gaits speed and swing foot placement. In the early development stage of the biped robots, a 17-DOF biped robot consisting 15 active DOF and 2 passive DOF was developed by Espiau et al. [83] under the French joint project BIP that achieved walking on flat terrains, inclined terrain, ascending, and descend- ing stairs. Since then, many approaches and models have been developed and shown their improved robustness. they reported that the sliding mode control performs better than the torque-based pure CTC technique to overcome the high nonlinearities of gaits on the stair- case. In this direction, Albert [84] developed a trunkless biped robot and designed a path planning mechanism to optimize nonlinearities. Some researchers have obtained stability by controlling the motion of the CoG of 7-DOF biped with large feet [85] and supervising of the ground center of pressure (GCoP) of 12-DOF biped by using the "hybrid-

state driven autonomous control (HyDAC)" algorithm. Besides that, researchers developed a mathematical model for interpolating third-order spline and monitoring the ZMP using MATLAB/SIMULINK. Later, Mandava and Vun- davilli

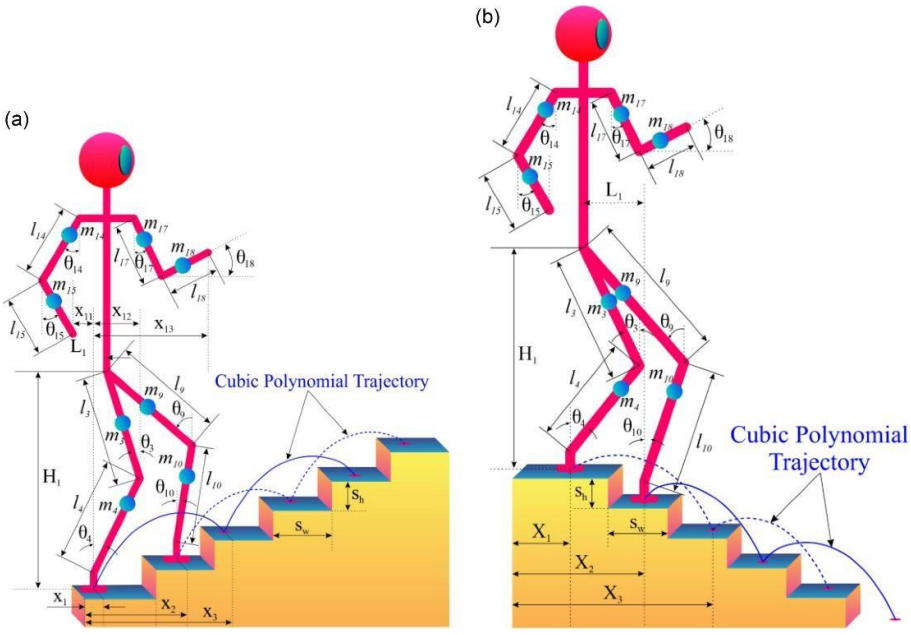

Fig. 8: Schematic diagram showing the gait generation of the biped robot (a) ascending the staircase and (b) descending the staircase

developed an optimal PID controller for an 18-DOF mini-sized humanoid robot and reported its better performance when optimized by a novel MCIWO algorithm than PSO. The developed algorithm also encompasses the deviations in slope inclination and staircase dimension. Researchers reported enhanced DBM due to cubic polynomial trajectory in swing foot and reduced hip height and increment in the height of the stair slope [86]. Bionic gaits consisting of AI-based approaches helped the researchers generate adaptive and autonomous gaits. Intelligence has been developed in biped models by implementing multilayered Hopfield kind NN, which resulted in autonomous trajectory generation [87]; architecture of building blocks comprising Reconfigurable Adaptive Motion Primitives (RAMPs); controller consisting of numerous neurons for energy efficient gait of NAO robot and managing small disturbances. FLC rule base optimized by GA and controller composed of NN and FLC. researchers reported that the MPSONN (Neural Network optimized by Modified Pthesis Swarm Optimization) required the least training time compared to MPSOFLC, PSONN, PSOFLC, and NN. And also, the they demonstrated better performance of the NN when optimized by MCIWO than differential evolution (DE) and PSO.

11.4 Gait generation for avoiding, crossing, and stepping over the obstacles

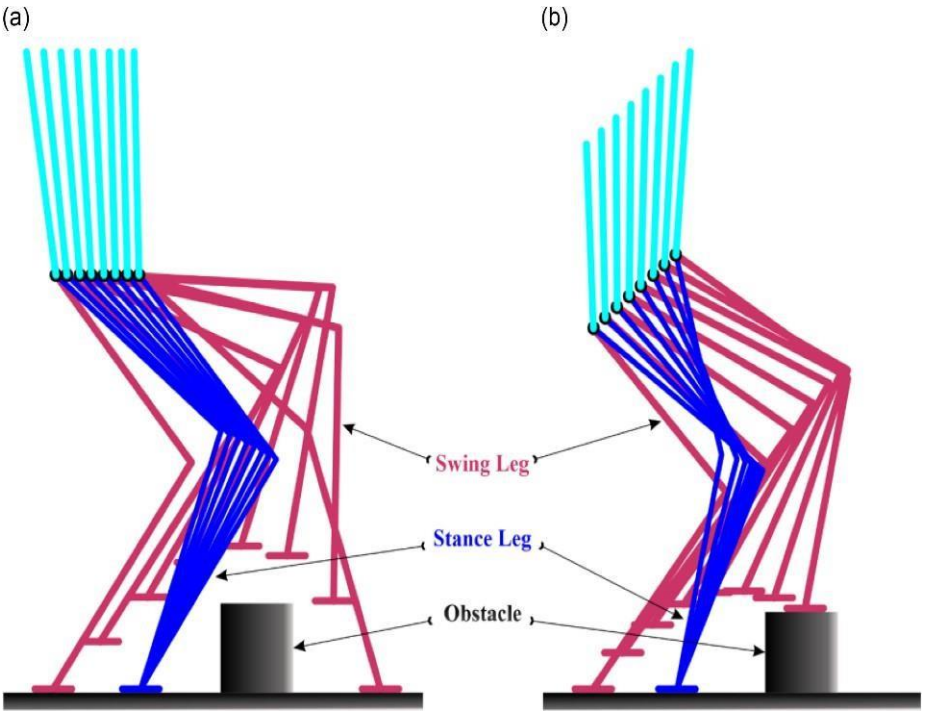

Fig. 9: Stick diagram showing the gait generation (a) crossing the obstacle, (b) stepping over the obstacle

The evolution of biped locomotion has the motive to develop a robust humanoid robot, efficient enough to perform all human motions. Humans inherit learnings from all sensory, intuitive knowledge, which is challenging to produce in the humanoid robot. But, applying some reinforced learning (RL) algorithms can develop intuitiveness in humanoid robots. To do so, many researchers have proposed some unique methodologies. Most of the researchers have shown interest in path planning to avoid obstacles. Very few have attempted to solve the problem of identifying the obstacles,and then crossing over or stepping over the obstacles. The perception-based control system was developed for generating walking primitive data of 16-DOF biped robot for step length adaptation, altering the direction and stepping over the obstacles. And also, the architecture consisting of the GA-NN and DE-NN that means NN trained by GA and DE, respectively, achieved the gait for crossing over the obstacles and positioning the foot on the obstacles as shown in Fig.9 along with generating the horizontal trajectory for hip and cubic polynomial trajectories for the swing foot respectively [88]. Gait while crossing the obstacles showed a more robust gait than positioning the foot on top of the obstacles. In addition to said methodologies, the self-navigation of biped robots has also been studied for a long time. Detecting the perception of the terrain is a very complex impediment for biped robot's navigation due to the limited view angle of visual sensors. That is why most of the navigation approaches are based on AI for the identification of the obstacles and then navigation around them for avoiding the obstacles. Such as, a novel hybridization framework consisting of a regression controller optimized with ant colony optimization (ACO); ZMP evaluation by using visual sensors; multi-modal sensory architecture having 6-DOF force-torque sensors at robot ankles and joint encoders for identifying the contact of the foot with a block [89];RA-FLC hybrid controller integrated with the Petri-net model and a control software consists of a stereo-camera driver; FL intelligent algorithm; integrated intelligence navigation controllers based on regression analysis and genetic algorithm approach for single and multiple NAO humanoid robots; a pure vision-based algorithm for the entire humanoid navigation strategy based on the topological map or visual memory (VM) by using an RGB-D camera [90] and 3D-SLAM (Simultaneous Localization and Mapping) by evaluating the next

viewpoint from a map through the camera for finding and holding the aimed object in unknown surroundings.

11.5 Gait generation for avoiding the dynamic obstacles

Many pieces of research have been carried out regarding the obstacles on the path of any biped robot. The proposed techniques and framework are efficient for avoiding, crossing, or stepping over stationary or static obstacles but do not consider moving or dynamic obstacles, representing a more realistic picture of walking in the natural environment. In this direction, Kashyap et al. [91] proposed an integrated DWA-TLBO (Dynamic-Window Approach and Teaching Learning Based Optimization) algorithm where positioning of target and obstacles are given to DWA as input for optimizing the speed and intermediate in-between consequences to TLBO and collectively evaluated optimum turning angle for avoiding the obstacles. The static navigation considers NAO, a mini-sized humanoid robot and stationary obstacles. In contrast, dynamic navigation con- siders several NAO robots where each NAO works as a dynamic obstacle for others with the help of a hybrid regression FL control approach. The researchers designed and applied a Petri-net controller in every NAOs to avoid clashing and validated the simulation and experiment results.

11.6 Gait generation for crossing over the ditches

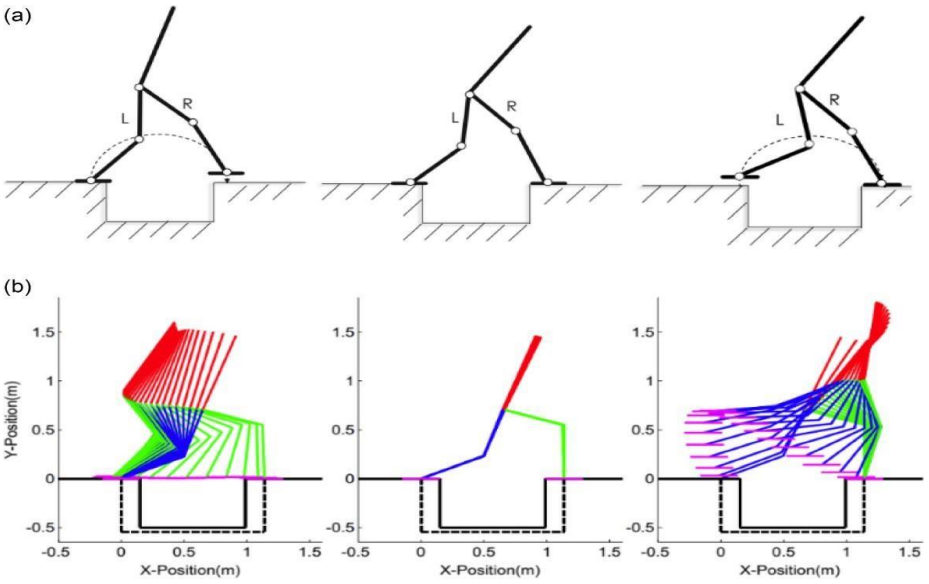

Fig. 10: . diagram of gait generation for crossing ditch (a) SSP, DSP SSP (left to right) phases of ditch crossing, (b) simulation of biped robot crossing ditch

The gait generation for crossing the ditches has been studied by only a few researchers, which is discussed in this section. Vundavilli and Pratihar [92] together proposed a gait planner for ditch crossing based on analytical modeling and two other techniques; NN and FL-based optimization of the dynamic balance margin and energy consumption for a 7-DOF biped robot. The NN and FL-based gait planners are trained offline by GA, enabling optimal online gait generation. The approaches other than the analytical modeling are more adaptive and more balanced for the minor energy consumption of a biped robot. In addition, they developed a multibody dynamics framework for gait genera- tion of 5-DOF biped robot as shown in Fig.10 (a), for giant steps and walking across wide ditches of width more significant than the leg length. The paths are

produced using time-independent constraints based on the distance trekked by COM. The approach is suitable for a robot similar to an adult human for going across the ditch of 1.05 m width with 0.2 lowest coefficients of friction. Later on, Janardhan and Kumar R [93] again proposed an approach for generating the dynamically balanced ditch crossing gait of width equal to or more than the length of the leg of a 7-DOF biped robot. The developed algorithm is incorporated with adopted constraints and adaptively tunes the time. Figure 10(b) shows that the simulation gave optimal joint torques and angle trajectories.the challenges and approaches for gait generation on uneven terrains have been presented in the next section.

11.7 Gait generation on the uneven terrains

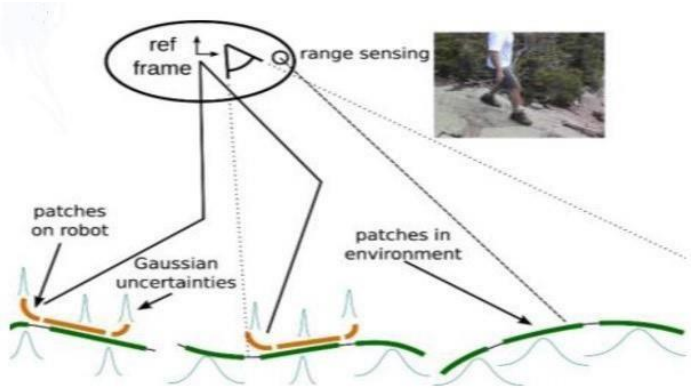

Fig. 11: the principle of the homogeneous patch map

Locomotion modeling on uneven terrain is challenging for modeling due to its uncertainties and not having specific patterns. That is why the foot placement is challenging to maintain dynamic balancing for biped robot walking on uneven or rough terrain. In this direction, worldwide researchers have developed LIPM-based biped model of massless legs [94]; LIPM-based simplified model for 42-DOF humanoid robot HRP-4C with ZMP delay; 3D LIPM-based 12-DOF model; algorithm for adopting 30 percent and 20 percent deviation in prescribed speed and step length respectively [95]; moving horizon technique for inheritance of human walking behavior on the INRIA designed biped robot "BIP"; versatile walk control framework by utilizing an ultra- sonic reach sensor for straight upset pendulum-based biped robot "Meltran-II" [96]; Poincare sections for asymptomatically stable periodic gait while regulating the impact of foot on the ground for an under- actuated biped robot; horizontally composed plane having unknown step height for a biped mechanism made up of viscous elastomer; a hybrid control consisting impedance control and CTC for swing leg and stance leg respectively with higher damping of leg while making contact with the ground and a robust adaptive controller inspired from "Turkey Walking" by virtual control for controlling speed, posture, and height Researchers developed an algorithm virtually with intuitive natural dynamics and applied it on Spring Turkey and Spring Flamingo based on a 7-link planar biped robot hav- ing contact switches on the foot. Furthermore, Manchester et al. [97] designed a controller by first making a lower-dimensional arrangement of directions cross- over to the objective cycle and then utilizing a subsiding skyline input regulator to dramatically balance out the linearized elements of the cross-over states rel- evant to HZD system and obtained the gait of non-periodic trajectories and switching over rough and irregular terrains. Addedly, Iida and Tedrake [98] employed open-loop sinusoidal oscillation of hip actuator and

developed a biped model of passive gait based on compass gait by changing the parameters of the oscillator.Intelligence-based gait generation techniques have been employed to improve the biped gaits' robustness on uneven terrain. researchers proposed hybrid intelligence methodology based on fuzzy NN controller and improved learning speed of any mobile robot to be controlled by itself on a real-time basis for sensing the direction of movement, target position by optical range finder and distances among various directions between obstacles with the help of ul- trasonic sensors in an unknown environment. In addition, they introduced a

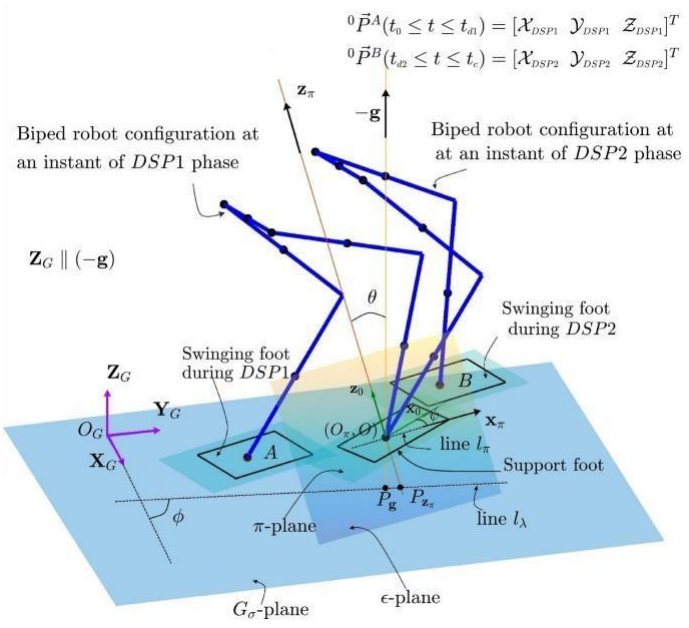

Fig. 12: bipedal robot on uneven trains

scheme for modeling, mapping, and tracking of rough rocky terrains for proper foot placement of robots on real-time data obtained from RGB-D and IMU sensor with the help of a set of parameterized patch models and bio-inspired sampling algorithms. The foot contacts are detected as bounded curved patches similar to foot sup- port consisting of sparse seed point sampling, point cloud neighborhood search, and patch fitting and validation [99]. The researchers also applied a 3D foothold perception architecture that utilizes the developed patch mapping and tracking scheme. In general, the dynamically stable robots fail to walk on slippery terrain; scientists suggest using moderate speed, short step lengths, and swing backward velocity. In addition, they generated online trajectories for CoM and ZMP by using the stability constraints with the help of an intrinsically stable MPC controller and applied it on the NAO. The pre- sented problems collectively can be termed as the unknown environment, as demonstrated in the next section [100].

11.8 Gait generation in the unknown environment

When modeling and mapping the exact perspective of the unknown or uncer- tain environment, it becomes difficult for a biped robot to make a quick decision based on observational and sensory data collected by various sensors and de- vices. The robot must have a quick decision-making framework that makes it an intelligence inbuilt mechanism. That is, more advanced technologies are re- quired for doing so. Along with the decision policy, its controller also needs to perform the basic controlling operations for maintaining the dynamic bal- ancing instantaneously. In comparing the various perspectives of intelligence in robotics and mechatronics, it can be said that animals are adaptable to their

environments, and humans make some changes in the environment for comfort [100]. That means basic intelligence is all about being adaptive to the dynamic environment and making some improvements in the environment is advanced intelligence. This section attempts to cover all perspectives and techniques proposed by various researchers around the globe in this direction. The bending of the knee joint at the lower hip position consumes more actuator torque. It was found that the minimum and maximum vertical distances between the ground and hip joint, and length of the shank and thigh greatly affect the torque [101]. The hip height has noble importance for stability, optimum actuator torques, preventing the link's velocity discontinuities, and deriving the lower torso's modified motion. Initially, a foot mechanism was configured by Yamaguchi et al. [102] for two biped robots WAF-3 and WL-12RVIII, to evaluate the relative position of foot support concerning the landing surface and the inclination of the ground.

Many researchers have proposed RL and training modules for interpolating intelligence and intuitive inheritance in bipedal walking robots. they used the inherited data from human locomotion and implemented all learnings to the radial basis function neural network (RBFNN) algorithm to generate the real-time gaits by using a visual system for making an autonomous humanoid robot based on a minimum energy principle. The results obtained from the GA and RBFNN were verified and compared by simulation on the humanoid robot "Bonten-Maru I." In addition to, the RL CPG actor-critic method learnt by policy gradient algorithm was implemented while introducing new schemes to the actor. Also, Pasandi et al. [103] discussed a CPG encompassing a novel bounded output oscillatory coherent network. Here, each oscillatory mechanism configures one dimensional intermittent function as a stable limit cycle. The CPG access the online trajectories library and generates the trajectory in real-time for the iCub humanoid robot. Later on, researchers developed a mathematical model to evaluate the path of the combined trajectory of a 7-DOF biped robot on different terrains. The effects of hip height on the torso's modified motions and then applied third-order spline was applied due to high accuracy and precision for determining the inverse kinematic, dynamics, and control variables [104] Similarly, Rioux and Suleiman [105] conjointly presented an entire navigation framework for a humanoid robot by creating a map of the environment and setting some primitives as a base knowledge for loading weights and avoiding obstacles without any sensors. The researcher presented an efficient filtering procedure to enhance the performance of the SLAM algorithm and clear the field by removing the cart from the view. The approach experimented on the NAO humanoid robot having an RGB-D sensor. In addition to this, Luo et al. [106] developed a real-time terrain realization sensory system by confining the vital hardware to a microprocessor and a single sort of force sensor and also investigated the gait pattern performance by grouping the SVM (Support Vector Machine) algorithm. Scholars observed that reinforcement learning, NN, CPG, mapping of the environment, and sensors-based systems had helped the researchers to make biped robots capable of walking in any unknown environment.

12 Chapter 3: Control methods

12.1 Overview of the Control Techniques

table 2 Overview of all the Control Techniques

Control Technique	Description	Strengths	Limitations
PID (Proportional-Integral-Derivative)	A feedback control system that calculates the error between desired and actual values, adjusting based on P, I, and D gains.	Simple, easy to implement, effective for general control tasks.	Struggles with large uncertainties (e.g., >80%) and nonlinear systems. Adaptive gain helps smooth operation, but less effective during complex phases like biped-in-air.
Neural Network (NN)	A control system with both offline and real-time learning capabilities, often integrated with Cerebellar Model Articulation Controllers (CMAC) for improved memory and learning speed.	Can adapt to changing environments, handles complex control tasks, improved learning speed with CMAC.	Computationally intensive, requires training data, and might struggle with very complex tasks without proper tuning.
Fuzzy Logic Controller (FLC)	Uses linguistic variables to handle imprecision in control tasks, adjusting for partial truths between 0 and 1.	Effective in dealing with nonlinearities, no need for precise mathematical models, good for complex or imprecise tasks.	May lack precision in highly structured environments; not ideal for tasks requiring exact models.
Computed Torque Controller (CTC)	Uses feedback linearization to simplify a nonlinear system into a linear one by eliminating cross-coupling and nonlinearities.	Useful for stabilizing dynamic systems, enhances trajectory tracking, can handle nonlinear systems when paired with linear controllers like PID or PD.	Requires an accurate dynamic model of the robot, limiting its applicability in real-time or uncertain environments.
Impedance Controller	Regulates the force and position of the robot's links by controlling the impedance, ensuring dynamic stability and robustness.	Versatile in handling various tasks, does not require solving inverse kinematics, adaptable to different environments.	Complex, may require tuning for specific environments and tasks.
Model Predictive Control (MPC)	Predicts future system behavior and computes optimal control actions based on dynamic optimization while respecting system constraints.	Handles complex multivariate systems, provides robust control against system parameter changes, effective for sophisticated control tasks.	Computationally expensive, requires a reliable model and real-time computation to adjust control actions.

12.1.1 the Design of the 3D bipedal robot

The bipedal model was developed in the Gazebo simulation environment, which can provide a high-performance 3-D physics simulation. The default physics engine used by GAZEBO is ODE (Open Dynamics Engine), which was used in this work. It also supports other physics engines, such as Bullet, Dynamic Animation and Robotics Toolkit (DART), and Simbody. GAZEBO [113] is popular in robotics and was used in the DARPA Robotics Challenge. It is often used with ROS (Robot Operating System), a popular API that provides commonly used tools for robotic applications. The simulated robot can be easily

controlled through the GAZEBO-ROS interface, as was done here. The model has a total of 12 joint DOF (degrees of freedom), 6 DOF on each leg, as shown in Figure.13: Three hip joint DOF, one knee joint, and two ankles joint DOF. Individual segment masses and lengths are proportioned based on human studies for a 1.8 m tall male. Mass and inertia are added to realistically represent an exoskeleton and its associated electronics as shown in Table 1. Joint limits are set to allow the full range of motion. The mass moment of inertia and the collision models are simplified as rectangular parallelepipeds. Mass and inertia properties for a particular user can be measured through experiments. The torso degrees of freedom are assumed fixed by the exoskeleton corset brace, as is done on the CPWS. In the GAZEBO environment we are going to test our bipedal robot balance in 3 different scenarios with 3 different control methods. first scenario is walking on the flat surface, second one is passing over a ditch and the third one is facing an obstacle. the PID, FLC and NN controllers are the controlling methods used in this research in order to see the efficiency and accuracy of each methods and compare their results with each other in the end. we especially chose these control methods because NN and FLC controllers are the most used controllers between other researchers and PID controller is the oldest control method which is also one of the common used controllers among the researchers. we wanted to compare the old control methods to the most recent ones and see their results. FIGURE.13.

Table 3. 3D biped robot technical specification

Link name	Mass (kg)	Dimension (mˆ3)	Offset (m)
Torso	50.85	0.36*0.18*0.72	(0,0,0)
Thigh	7.5	0.1*0.1*0.441	(0,0,0)
Shank	4.4875	0.1*0.1*0.414	(0,0,0)
Foot	1.5	0.11*0.17*0.03	(0,0,0)
Backpack	5	0.36*0.1*0.18	(0,0.1,0)
Exoskeleton thigh	2	0.1*0.1*0.441	(0,0.05,0)
Exoskeleton shank	2	0.1*0.1*0.414	(0,0.05,0)

13 controlling approaches

13.1 PID controller

The PID controller, also known as a proportional–integral–derivative controller or three-term controller, serves as a feedback mechanism extensively utilized in industrial control systems and various other contexts necessitating continuous modulation of control. Its operation involves the continuous computation of an error value, represented as e(t), reflecting the disparity between a desired set point (SP) and a measured process variable (PV). The controller then applies corrections based on proportional (P), integral (I), and derivative(D) terms, hence its name [114]. PID systems autonomously administer precise and prompt

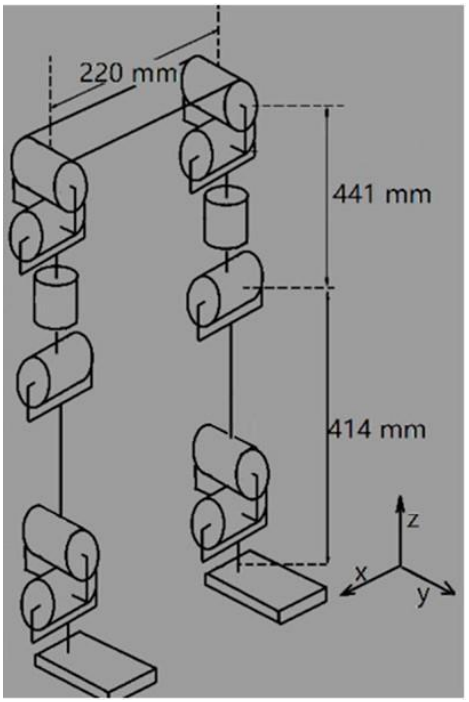

Fig. 13: GAZEBO simulation model. Each leg has six degrees of freedom: Three at the hip, one at the knee, and two at the ankle.

adjustments to a control function. A common instance is the cruise control feature in automobiles, where ascending a hill would otherwise reduce speed if constant engine power were maintained. The PID algorithm of the controller promptly restores the measured speed to the desired level, mitigating delay and overshoot by systematically enhancing the engine's power output. The initial theoretical analysis and practical application of PID originated in the domain of automatic steering systems for ships, evolving from the early 1920s onwards [115]. Subsequently, it found widespread adoption in the manufacturing industry for automatic process control, initially in pneumatic systems and later in electronic controllers. Presently, the PID concept enjoys universal utilization across applications necessitating precise and optimized automatic control.

13.1.1 PID control on a flat surface

The present work is focused on the design of a suitable PID controller that can generate dynamically balanced gaits for the 12-DOF biped robot on a flat surface. Once the gait is generated, the dynamic balance margin of the biped robot is verified by using the concept of zero moment point (ZMP)[116]. Further, the parameters such as linear and angular velocity of the joints are very important for locomotion of the biped robot. It is important to note that Jacobian is used

to determine the linear(Jvi) and angular velocity (Jwi) of the joints of the biped robot also the Zi−1:I in eq (5) is The vector along the axis of the previous joint (frame or axis) [117].

$$J_i = \begin{matrix} J_{vi} \\ J_{\omega i} \end{matrix} = \begin{matrix} Z_{i-1} \times (O_n - O_{i-1}) \\ Z_{i-1} \end{matrix} \quad (5)$$

The dynamics of the biped robot is determined with the help of Lagrange–Euler formulation, which is given below.

$$\tau_{i,the} = \sum_{j=1}^{n} M_{ij}(q)\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} C_{ijk}\dot{q}_j\dot{q}_k + G_i \quad i,j,k = 1,2\ldots n(6)$$

where Ti indicates the theoretical torque required at joint i(N-m), q, qj and represent the displacement, velocity (rad/s) and acceleration (rad/sec2) of the joint, respectively. Further, the expanded form of inertia term (Mij), Coriolis/centrifugal forces (Cijk) and gravity terms (Gi) are given below[118].

$$M_{i,j} = \sum_{p=max(i,j)}^{n} Tr\ d_{pj} I_p d_{pi}^T \quad i,j = 1,2\ldots n(7)$$

$$C_{i,j,k} = \sum_{p=max(i,j,k)}^{n} Tr\ \frac{\partial(d_{pk})}{\partial q_p} I_p d_{pi}^T \quad i,j = 1,2\ldots n(8)$$

$$G_i = -\sum_{p=i}^{n} m_p g d_{pi}^p \overline{r_p} \quad i,j = 1,2\ldots\ldots\ldots\ldots n(9)$$

where Ip

represents mass moment of inertia (kg-m/s2) tensor and mass center (m) of pth link, respectively and g denotes acceleration due to gravity in (m/s2). While controlling the joint, the acceleration of the link plays a major role. Therefore, the expression in terms of acceleration will be obtained by rearranging the Eq. (10), and is given below.

$$\ddot{q}_y = \sum_{j=1}^{n} M_{ij}^{-1}(q)\left[\tau_{i,\text{the}} - \sum_{j=1}^{n}\sum_{k=1}^{n} C_{ijk} q_y \dot{q}_k - G_i\right] \quad (10)$$

Now by considering the term

$$\sum_{j=1}^{n} M_{ij}(q)^{-1} \times \tau_{i,the} = \hat{\tau}(11)$$

13.1.2 Design of PID controller for the biped robot

The Proportional-Integral Derivative (PID) controllers are widely used to control the motors of the biped robot in various applications. The expression for the joint based PID controller that is used in this study is mentioned below [119].

$$\tau_{act} = K_p e + K_d \dot{e} + K_i \int edt(12)$$

where Kp , Kd and Ki represents proportional, derivative and integral gains of the controllers, respectively. The expanded form of the above equation after including the meaning is given below.

$$[e(\theta_i) = \theta_{if} - \theta_{is}]$$
$$\tau_{i,\text{act}} = [K_{pi}(\theta_{if} - \theta_{is}) - K_{di}\theta_{is} + K_{ii}\int e(\theta_{is})\ dt] \quad i = 1,2,\dots,n \tag{13}$$

where Ti,act represent the actual torque supplied by the controller to individual joints to move from an initial angular position to final angular position . Further, the integral terms in the above equation need to be substituted by its state variables, namely and its meaning is given below.

$$x_i = \int e(\theta_{is})\ dt \quad \Rightarrow \quad \dot{x}_\iota = \theta_{if} - \theta_{is} \quad i = 1,2,\dots,n \tag{14}$$

The final control equation that represents the control action for all joints is given below.

$$q_j = \sum_{j=1}^{n} M_{ij}(q)^{-1}\left[-\sum_{j=1}^{n}\sum_{k=1}^{n} C_{ijk}q_y\dot{q}_k - G_i\right] + K_{pi}(\theta_{if} - \theta_{is}) - K_{di}\theta_{is} + K_{ii}\int e(\theta_{is})\ dt \tag{15}$$

PID Controller Pseudo Code

```
START
1. Initialize setpoint (SP), initial process value (PV), and gains (Kp, Ki, Kd).
2. Set previous error (prev_error) = 0 and integral = 0.
3. Loop until the system stabilizes:
    a. Calculate error = SP - PV.
    b. Calculate integral = integral + error * dt.
    c. Calculate derivative = (error - prev_error) / dt.
    d. Calculate output = (Kp * error) + (Ki * integral) + (Kd * derivative).
    e. Update system using the output.
    f. Set prev_error = error.
    g. Update PV from the system.
4. End loop.
END
```

13.2 Formulation as an optimization problem

The design of controller for the biped robot to achieve a dynamically balanced gait on the flat surface can be considered as an optimization problem as explained below. The controllers have to supply the torque that is required by the joint to move from an initial position to final position. Further, the torque supplied by the controller will be able to reduce the positioning error with certain limits on the range of the gain values of the controller. Moreover, the positional error related to the both swing leg and stance leg is also to be considered [120]. Then it may be posed as an optimization problem as given below:

For Swing Leg

Minimize Z1: $e_1 = \sum_{i=1}^{6} e(\theta_i)$(16)

Subjected to constraints: $K_{pi,min} \le K_{pi} \le K_{pi,max}$

$K_{di,min} \le K_{di} \le K_{di,max}$

$K_{ii,min} \le K_{ii} \le K_{ii,max} \quad i = 1, 2, \ldots 6.$

For Stand Leg

Minimize Z2: $e_2 = \sum_{j=7}^{12} e(\theta_j)$(17)

Subjected to constraints: $K_{pj,min} \le K_{pj} \le K_{pj,max}$

$K_{dj,min} \le K_{dj} \le K_{dj,max}$

$K_{ij,min} \le K_{ij} \le K_{ij,max} \quad j = 7, 8, \ldots 12.$

where e1 and e2 are the total error in the angular position of all the joints of the swing and stand leg, respectively. The e(0i) and e(0j) are the error at the individual joints of the both legs and rest of the terms carries their usual meaning.

13.2.1 Proposed optimization algorithm

in the present work, a non-traditional optimization algorithm, namely MCIWO is used to tune the gains of the PID controller and compared with another established nature inspired optimization algorithm such as, PSO [121].

13.3 Modified chaotic invasive weed optimization algorithm

Invasive weed optimization (IWO) is a stochastic optimization algorithm developed by Mehrabain and Lucas in 2006 [122] and the algorithm is inspired from the colonizing behavior of the weeds. In a crop field the weeds are randomly dispersed and live in between them. After randomly placing these weeds, they take the unused resources in the cropping field and grows to a flowering weed plant and produce new seeds.

The number of seeds produced by each flowering weed depends on the fitness of each flowering weed plant. These seeds will develop new weeds in the field, and they have better adaption in the environment and take more resources that are unused in the field. They grow very fast and produce more number of seeds from each weed plant. This process will continue until the maximum number of weeds grow in a field by using the limited resources. Initially, the seeds that represent the N- dimensional solution space of the problem will be generated at random [123].

the gains of the PID controller are considered as the solution space of the problem. Once the initial seeds grow into flowering plants after using the unused resources, the fitness of each plant will be evaluated. In the present problem, the fitness in terms of average angular positional error of the PID controllers are considered as the fitness of the plant[124]. Once the fitness is determined, the process of reproduction starts to determine the number of seeds produced by the plant. The equation that represents the reproduction scheme is given in Eq. (18).

$$S=\text{Floor}\left(S_{min}+\frac{f-f^{min}}{f_{max}-f_{min}}\times S_{max}\right) \quad (18)$$

where (f min , f max) and (S min , S max) denotes the minimum and maximum fitness of the colony and minimum and maximum seeds produced by the plant, respectively. Then the newly produced seeds are randomly distributed using normal distribution with mean zero and variance in the search space.

To improve the performance of the algorithm, in the present work two terms namely chaotic variable and cosine variable are added during reproduction. Here, the function of chaotic variable is to minimize the chances of the solution to trap in the local optimum point by increasing the spread of the new seed dispersion area. It is derived by using chebyshev map and is represented by the expression given below.

$$X_{k+1}=\cos\left(k\cos^{-1}(X_k)\right) \quad (19)$$

Moreover, the value of standard deviation that is used to define the position of the seed is given as follows.

$$\sigma_{Gen} = \frac{(Gen^{max} - Gen)^{n}}{(Gen_{max})^{n}} \times |\cos(Gen)| \times (\sigma_{initial} - \sigma_{final}) + \sigma_{final} \quad (20)$$

where Gen max is the maximum number of generations, n is the modulation index i.e. integer number, o initial and o final is the initial and final value of the standard deviation. The term [cos(Gen)] not only helps in determining the global optimal solution, but also enhances the search space by utilizing the minimum resources. It can be observed that after modification, the algorithm explores more search space compared to standard IWO algorithm. Once the new seeds are obtained from the reproduction scheme, their fitness is also being evaluated and ranked along with the parent plants. It is important to note that the maximum number of plants should not exceed Pmax. Then competitive exclusion is performed to remove the lower ranked plants. This process will continue until the maximum number of iterations are reached [125].

14 Gait generation while facing an obstacle

The motion of the biped robot is considered in both the sagittal as well as frontal planes of the biped robot. The orientation of each joint of the biped robot has been calculated by using the concept of inverse kinematics. The mathematical expressions that are used to determine the joint angles of the swing leg (i.e., θ_3 andθ_4) and stand leg (i.e., θ_9 and θ_{10}) in sagittal plane, and the joint angles of swing leg (i.e., θ_2 and θ_5) and stand leg (i.e., θ_8 and θ_{11}) in frontal plane are summarized below.

θ_4 = sin1 (H1l3 sin ψ +$L1l3cos\psi(l4+l3cos\psi)(l4+l3cos\psi)2+(l3sin\psi)2)(21)$

where L1 = l4 sin(θ_4) + l3 sin(θ_3), H1=l4 cos(θ_4)+l3 cos(θ_3),ψ =θ_4 θ_3 = arcos (H1 2 + L1 2 l4 2 l3 2 2l4l3) (22)

Thus, θ_3 can be obtained from the equation $\theta_3=\theta_4 -\psi.Similarly, the angles\theta_9$ and θ_{10} are obtained by using the following mathematical expressions.

θ_{10} = sin1 (H2l9 sin ψ + $L2(l10 + l9cos\psi)/(l10 + l9cos\psi)2 + (l9sin\psi)2)(23)$

where L2 = l9 sin(θ_9) + l10 sin(θ_{10}), H2 = l9 cos(θ_9) + l10 cos(θ_{10}), =θ_{10} θ_9 =arcos (H2 2+L2 2 l10 2 l9 2/ 2l10l9) (24)

Thus, θ_9 can be obtained from the equation $\theta_9=\theta_{10} - \Psi(25)$

. The included angle at the ankle joint with respect to the vertical axis is assumed to be equal to zero i.e., $\theta_6=\theta_{12}$=0. For determining the joint angles of the leg in frontal plane, the following equations are considered.

= θ_8 = tan1 (fw/ H1) θ_5 = θ_{11} = tan1 (0.5fw/ H2) (26)

The following equations are used to determine the position of ZMP in X-direction and Y-direction.

$$x_{ZMP} = \frac{\sum_{i=1}^{n} (I_i \cdot \theta_i \cdot m_i \cdot \ddot{x}_i \cdot z_i + m_i \cdot x_i \cdot (g \cdot z_i))}{\sum_{i=1}^{n} m_i \cdot (\ddot{z}_i \cdot g)} \quad (27)$$

$$y_{ZMP} = \frac{\sum_{i=1}^{n} (I_i \cdot \theta_i \cdot m_i \cdot \ddot{y}_i \cdot z_i + m_i \cdot y_i \cdot (g \cdot z_i))}{\sum_{i=1}^{n} m_i \cdot (\ddot{z}_i \cdot g)} \quad (28)$$

where w denotes the angular acceleration (rad/s2), Ii is mass moment of inertia (kg·m2) of the i-th link, g is the acceleration due to gravity (m/s2), mi represents the mass (kg) of the link i, and and indicate the acceleration of the i-th link moving in z and x directions (m/s2) and (xi, yi, zi) signifies the coordinates of the i-th lumped mass. After determining the zero moment point, the dynamic balance margin has been calculated in both X-direction and Y-direction . The dynamic balance margin has been defined as the difference between the length of the foot to the point where ZMP is acting on the foot support polygon. It is interesting to note that while walking on the stair case, the foot of the robot is always parallel to the stair width.

The gait generation method for the biped robot while descending the obstacle is similar to the ascending case except for a small difference in determining the position of ZMP. While descending the stair case, the acceleration due to gravity (g) is acting in the direction opposite to that of the movement of robot. The expressions for ZMP in both X-direction and Y-direction while descending the stair case are given below [126].

$$x_{ZMP} = \frac{\sum_{i=1}^{n} (I_i \cdot \theta_i \cdot m_i \cdot \ddot{x}_i \cdot z_i + m_i \cdot x_i \cdot (g \cdot z_i))}{\sum_{i=1}^{n} m_i \cdot (\ddot{z}_i \cdot g)} \quad (29)$$

$$y_{ZMP} = \frac{\sum_{i=1}^{n} (I_i \cdot \theta_i \cdot m_i \cdot \ddot{y}_i \cdot z_i + m_i \cdot y_i \cdot (g \cdot z_i))}{\sum_{i=1}^{n} m_i \cdot (\ddot{z}_i \cdot g)} \quad (30)$$

The torque based PID controllers are designed for each joint of the biped robot while moving on different terrain conditions. For designing a torque based PID controller initially, the dynamics of the biped robot is to be derived. here we used Lagrange-Euler formulation for calculating the dynamics of the biped robot.

$$\tau i, the = nj = 1Mij(q)..qj + nj = 1nk = 1Cijk.qj.qk + Gii, j, k = 1, 2, , n. (31)$$

Furthermore, the acceleration of the links plays a major role in controlling each joint of the biped robot. Therefore, the expression in terms of acceleration of the link can be obtained by rearranging the above equation as

$$qj = n\ j=1\ Mij\ (q)\ 1\ [\ n\ j=1\ n\ k=1\ Cijk\ .\ qj\ .\ qk\ Gi\ ] + (n\ j=1\ Mij\ (q)\ 1 \times \tau i, the), i, j, k = 1, 2, , n (32)$$

where the terms indicate the theoretical torque required for each joint i (N·m), denotes the displacement in (rad), represent the velocity in (rad/s), qs is the acceleration in (rad/s2) of the joint, respectively. Now consider the term:

$$n\ j=1\ Mij\ (q)\ 1 \times \tau i, the = (33)$$

After substituting the above term in (32), it can be written as

$$qj = n\ j=1\ Mij\ (q)\ 1\ [\ n\ j=1\ n\ k=1\ Cijk\ .\ qj\ .\ qk\ Gi\ ] + \tau i, j, k = 1, 2, , n (34)$$

The actual torque required at different joints of the biped robot after utilizing the torque based PID controller is calculated by using the following expression

$$\tau act = Kpe + Kd.e + Kiedt (35)$$

where t act indicates the actual torque required at each joint and , and represents proportional, derivative and integral gains of the controller,respectively. The expanded form of the above equation after including the meaning of e is given below

$$e\ (\theta_i) = \theta_{if}\,\theta_{is}\tau i, act = Kpi(\theta_{if}\,\theta_{is})Kdi.\theta_{is} + Kiie(\theta_{is})dt, i = 1, 2, , n (36)$$

where represents the actual torque supplied by the controller to individual joints to move from an initial angular position to final angular position . The final control equation that represents the control action for all joints is given below.

qj = n j=1 Mij (q) 1 [n j=1 n k=1 Cijk · qj · qk Gi] + Kpi ($\theta_{if}\theta_{is}$)$Kdi\theta_{is}$ + $Kiie(\theta_{is})$dt.(37)

Once the PID controller is designed for each joint of the biped robot, then tuning of the gains of the PID controller is posed as an unconstrained optimization problem. It is to be noted that the positional error related to both the swing and stance leg is considered as an objective function for the optimization problem as explained below.

Minimize Z = $\sigma_6 i$ = 1$e(\theta_i)$ + $\sigma_i j$ = 7$e(\theta_j)$.(38)

Subjected to constraints

Kpi,min Kpi Kpi,max

Kdi,min Kdi Kdi,max

Kii,min Kii Kii,max

i = 1, 2, · · · , 6

Kpj,min Kpj Kpj,max

Kdj,min Kdj Kdj,max

Kij,min Kij Kij,max

j = 7, 8, · · · , 12

where Kp, Kd and Ki are having their usual meanings and i and j represent the joints of swing and stance leg, respectively.

15 Passing over a ditch

15.1 Dynamics Equations

Dynamics Modeling

One walking step may include two phases, the single support phase (SSP), when one leg is virtually pivoted to the ground while the other leg is swinging in the forward direction (open kinematics chain configuration) and the double support phase (DSP), when both legs remain in contact with the ground while the entire system is swinging in the forward direction (closed kinematics chain configuration).

Having θ_i, as the generalized coordinates, the dynamic equations of motions of the biped can be derived in a simpler form. Dynamic equations of the biped robot shown are derived using Lagrange formulation and can be expressed as:

$$\mathbf{M}(\theta)\ddot{\theta} + \mathbf{V}(\theta)\mathbf{u} + \mathbf{G}(\theta) = \mathbf{T}_\theta = \mathbf{D}\tau (39)$$

where θ is the 6 × 1 joint variable vector, $\mathbf{M}(\theta)$ is the 6 × 6 symmetric inertia matrix, $\mathbf{V}(\theta)\mathbf{u}$ is the 6 × 1 vector of Coriolis and centripetal torques, $\mathbf{G}(\theta)$ is the 6 × 1 vector representing gravitational torques, $\mathbf{T}_\theta$ is the 6 × 1 vector of generalized torques that corresponds to θ and τ is the 6 × 1 vector of control input torques of the six joints of the biped.

1) Dynamics Equations in SSP:

This phase begins with the foot of the swing leg leaving the ground and terminates with the swing foot touching the ground. The contact point between the stance foot and the ground is fixed in the fixed frame and the friction between the foot and the ground is assumed to be sufficient to prevent slippage during

walking. Hence, the robot has 6 DOF in this phase and general form of dynamic equations of the motion can be written with elimination of the last row and column of $\mathbf{M}(\theta)$ and $\mathbf{V}(\theta)$ and also elimination of the last row of $\mathbf{D}$, $\mathbf{G}(\theta)$, $\mathbf{u}$, and θ, in (39), namely:

$$\mathbf{M}_{6\times6}\ddot{\theta} + \mathbf{V}_{6\times6}\mathbf{u} + \mathbf{G}_{6\times1} = \mathbf{D}_{6\times6}\tau \quad (40)$$

2) Dynamics Equations in DSP:

In this phase, both feet are in contact with the ground while the body can move forward slightly. As both of the contact points between the feet and the ground are fixed during the DSP, there exist two holonomic constraint equations for walking on level ground as follows:

$$\phi(\theta) = (x_h - x_t - L)(z_h - z_t) = 0 \quad (41)$$

where (x_t, Z_t) is position of the tip of rear foot, (x_h, Z_h) is position of the heel of front foot and L is distance between this two contact points in x-axis direction.

In (39), the vector of generalized coordinates is 6×1 and there are two constraints and thus the robot has 6 DOF in this phase. With the definition of λ as the 2×1 vector of Lagrange multipliers and $\mathbf{J} = \partial\Phi/\partial\theta$ as 2×6 Jacobian matrix, dynamic equations of the motion in this phase can be expressed as:

$$\begin{aligned} &\mathbf{M}(\theta)\ddot{\theta} + \mathbf{V}(\theta)\mathbf{u} + \mathbf{G}(\theta) = \mathbf{D}\tau + \mathbf{J}^{\mathbf{T}}(\theta)\lambda \\ &\varphi(\theta) = \mathbf{o} \end{aligned} \quad (42)$$

These two set of the equations must be solved simultaneously which is not an easy task. Upon definition of $\mathbf{J}_c$ as orthogonal compliment of $\mathbf{J}$ namely $\mathbf{J}_c = \mathbf{o}$, equations of motion can be expressed as:

$$\bar{\bar{\mathbf{M}}}_{6\times6}\ddot{\theta} + \bar{\bar{\mathbf{N}}}_{6\times1} = \bar{\bar{\mathbf{D}}}_{6\times5}\tau \quad (43)$$

where

$$\bar{\mathbf{M}} = \begin{matrix}\mathbf{I}\\ \mathbf{J}\end{matrix}\begin{matrix}\mathbf{J}_c^{\mathbf{T}}\mathbf{M}\\ \end{matrix}, \bar{\mathbf{N}} = \begin{matrix}\mathbf{I}\\ \dot{\mathbf{J}}\dot{\theta}\end{matrix}\begin{matrix}\mathbf{J}_c^{\mathbf{T}}(\mathbf{V}\mathbf{u} + \mathbf{G})\\ \end{matrix}, \bar{\mathbf{D}} = \begin{matrix}\mathbf{I}\\ \mathbf{o}\end{matrix}\begin{matrix}\mathbf{J}_c^{\mathbf{T}}\mathbf{D}\\ \end{matrix} \quad (44)$$

16 STABILITY CRITERION

We use the ZMP criterion which was first introduced by Vukobratovic [126] to evaluate the stability of the humanoid robot. The ZMP is defined as the point on the ground where the tipping moment acting on the biped, due to gravity and inertia forces, equals zero. The convex hull of the contact points between the feet and ground are called stable region. If the ZMP is within the stable region, the robot can walk [127]. The minimum distance between the ZMP and the boundary of the stable region is called the stability margin. The larger the stability margin results in the higher stability of the robot. The ZMP in x-axis and y-axis directions can be computed using the following equations:

$$x_{ZMP} = \frac{\sum_{i=1}^{n} \{m_i(\ddot{z}_i - g_z)x_i - m_i(\ddot{x}_i - g_x)z_i - I_y\alpha_y\}}{\sum_{i=1}^{n} m_i(\ddot{z}_i - g_z)} \quad (45)$$

$$y_{ZMP} = \frac{\sum_{i=1}^{n} \{m_i(\ddot{z}_i - g_z)y_i - m_i(\ddot{y}_i - g_y)z_i + I_x\alpha_x\}}{\sum_{i=1}^{n} m_i(\ddot{z}_i - g_z)} \quad (46)$$

where n is the number of links, m_i is the mass of links, I_{xi} and I_{yi} are the inertial component of links, α_{xi} and α_{yi} are the absolute angular acceleration components around x-axis and y axis at the center of gravity of links respectively, (g_x, g_y, g_z) is the vector of gravitational acceleration, (x_i, y_i, z_i) is the coordinate of the center of mass of links in an absolute Cartesian coordinate system and $(x_{ZMP}, y_{ZMP}, 0)$ is the coordinate of the ZMP.

16.0.1 TRAJECTORY PLANNING

Each leg trajectory in sagittal plane can be determined by a vector as $\mathbf{V}_a(t) = [x_a(t), z_a(t), \theta_a(t)]$ where denotes the ankle joint position and the angle of the sole about the ground and a vector as $\mathbf{V}_h(t) = [x_h(t), z_h(t), \theta_h(t)]$ where denotes the hip joint position and the angle of the trunk about the vertical line. If hip and ankle joints trajectories of each leg are known, all other joint trajectories will be determined using inverse kinematics. We only discuss trajectories in the sagittal plane but the hip abduction/adduction motion can be obtained similar to the sagittal hip motion.

Each walking step includes two phases of SSP and DSP. Interval of the DSP should be determined carefully. If we assume that the DSP is instantaneous, the hip has to move too fast in order to maintain its stability, namely ZMP must be transferred from the rear foot to the front foot during the short interval. On the other hand, if we assume that the interval of the DSP is too long, it is difficult for the biped robot to walk at high speed. The interval of the DSP in human walking is about 20% [128], so we used this value for our calculation. Assuming that the interval of one walking step is T_c and the interval of DSP is T_d. As mentioned before, we assume $T_d = 0.2\ T_c$. We define that a walking step starts with the heel of one of the feet leaving the ground at $t = 0$ and finishes with the

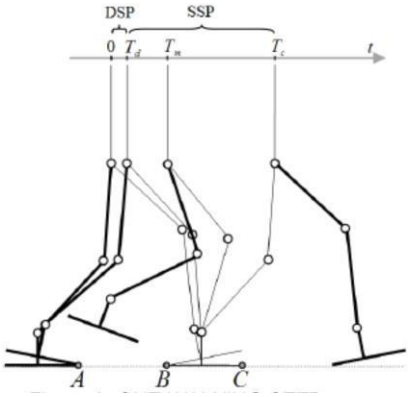

Fig. 14: one walking step

heel of the same foot making first contact with the ground at t = T_c, as shown in Fig. 14.

Since walking is a periodic phenomenon, trajectory of one of the legs can be used for another leg except with a T_c delay. Hence, we discuss only the generation of the right leg trajectory in a complete walking cycle. The left leg trajectory is same as the right foot trajectory except with a T_c delay.

16.1 Foot Trajectory

In most studies, it is assumed that the foot angle with the ground remains zero during the walking. Since the robot can not touch the ground first by the heel of the forward foot and leave the ground finally by the toe of the rear foot, these kinds of foot trajectories are not useful for high-speed walking. So, let say q_b and q_f be the angles of the right foot as it leaves and lands on the ground, respectively. We suppose that biped walks on a flat ground, so the foot angle constraints in a complete walking cycle can be expressed as:

$$\theta_{ra}(t) = \theta_1(t) = \{ 0 \; t = 011 q_b t = T_d q_f t = T_c 0 T_c + T_d \le t \le 2T_c (47)$$

To eliminate the impact effect of feet with ground, having the lowest energy consumption in ankle actuators as well as continuity of $\theta_{ra}(t)$ and ZMP in all walking steps, the following constraints must be satisfied:

$$\{ \dot{\theta}_{ra}(t) = 0, \ddot{\theta}_{ra}(t) = 0 t = 0$$
$$\dot{\theta}_{ra}(t) = 0 t = T_d$$
$$\dot{\theta}_{ra}(t) = 0 t = T_c$$
$$\dot{\theta}_{ra}(t) = 0, \ddot{\theta}_{ra}(t) = 0 t = T_c + T_d \; (48)$$

In environments with obstacles, it is necessary to lift the swing foot high enough to avoid obstacles. To this end, the following constraints must be satisfied:

$x_{ra}(t) = \{ l_f (1 - \cos\theta_{ra}) + l_a \sin\theta_{ra} 0 \le t \le T_d \; L_{ao} t$
$= T_m$
$2D_s - l_b (1 - \cos\theta_{ra}) - l_a \sin\theta_{ra} T_c \le t \le 2T_c \; z_{ra}(t) = \{ l_f \sin\theta_{ra} + l_a \cos\theta_{ra} +$
$h_{gs} 0 \le t \le T_d$
$H_{ao} t = T_m$
$l_b \sin\theta_{ra} + l_a \cos\theta_{ra} + h_{ge} T_c \le t \le 2T_c$ (49)

Here, (L_{ao}, H_{ao}) is the position of the highest point of the swing foot. Also D_s is the length of one step, T_m is the time when the right foot is at its highest position, $1_1 = l_7 = l_a$ is the height of the foot, l_f is the length from the ankle joint to the toe, 1_b is the length from the ankle joint to the heel. Moreover, h_{gs} and h_{ge} are the distances between the ground surface which is under the support foot (left foot) and the ground surface which is under the swing foot (right foot) at the start and end of the SSP, respectively. Since right foot ankle joint is at its highest position at T_m, the first time derivative of $z_{ra}(t)$ at T_m must be zero.

16.1.1 Hip Trajectory

It is desirable when there is no waist joint, trunk angle stays constant during walking namely, $\theta_h(t) = 0$. Studies show that the variation of hip height hardly affects the position of the ZMP. Also, to reduce the load on the knee joint, it is essential to keep the hip at a high constant position [129]. Hence, we specify hip height $z_h(t)$ to be constant during the walking.

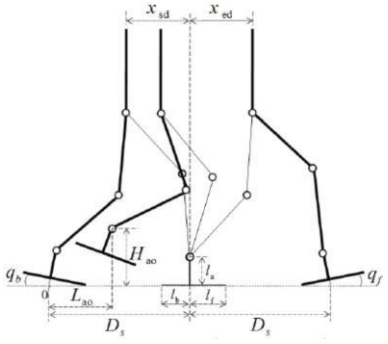

Fig. 15: walking parameters

The change of hip position in x-axis direction $x_h(t)$ is the main factor that affects the stability of a biped robot during walking in the sagittal plane. Let x_{sd} and x_{ed} denote distances along the x-axis from the hip to the ankle of the support foot at the start and end of the SSP, respectively, as shown in Fig.15. So, using Fig.14 and Fig.15 we have the following constraints:

$x_h(t) = \{ x_{ed} \; t = 0$
$D_s - x_{sd} t = T_d$

$D_s + x_{ed} t = T_c$ (50)

To obtain smooth and periodic trajectories for hip motion, the following constraints must be satisfied:

$$\dot{x}_h(t=0) = \dot{x}_h(t=T_c) \quad (51)$$
$$\ddot{x}_h(t=0) = \ddot{x}_h(t=T_c)$$

To generate smooth trajectories and continuous ZMP, it is necessary that the velocity and acceleration terms of foot and hip joint trajectories be continuous all the times. So with having the kinematics constraints for foot and hip joints, we can generate smooth trajectories for foot and hip joints using polynomials with suitable orders such that the first and the second time derivatives are continuous all the times.

By varying the values of walking parameters q_b, q_f, L_a, H_a, x_{sd} and x_{ed}, different foot and hip trajectories can be easily produced but the trajectories that guarantee the dynamic stability of robot during the walking, must be selected. As mentioned, two parameters of hip motion in x-axis direction x_{sd} and x_{ed} have the most effect on the stability of a biped robot in the sagittal plane. Therefore, we plan a method to find only these two parameters so that guarantees the dynamic stability of robot during the walking [129].

Stable region in a walking step is shown in Fig.16. In order to stable waking, the ZMP must cross A, B, and C points that are also shown in Fig.16. Since walking is a periodic phenomenon, the state of ZMP in A and C points are the same. Therefore, if the ZMP only cross A and B points, the biped is able to have a stable motion.

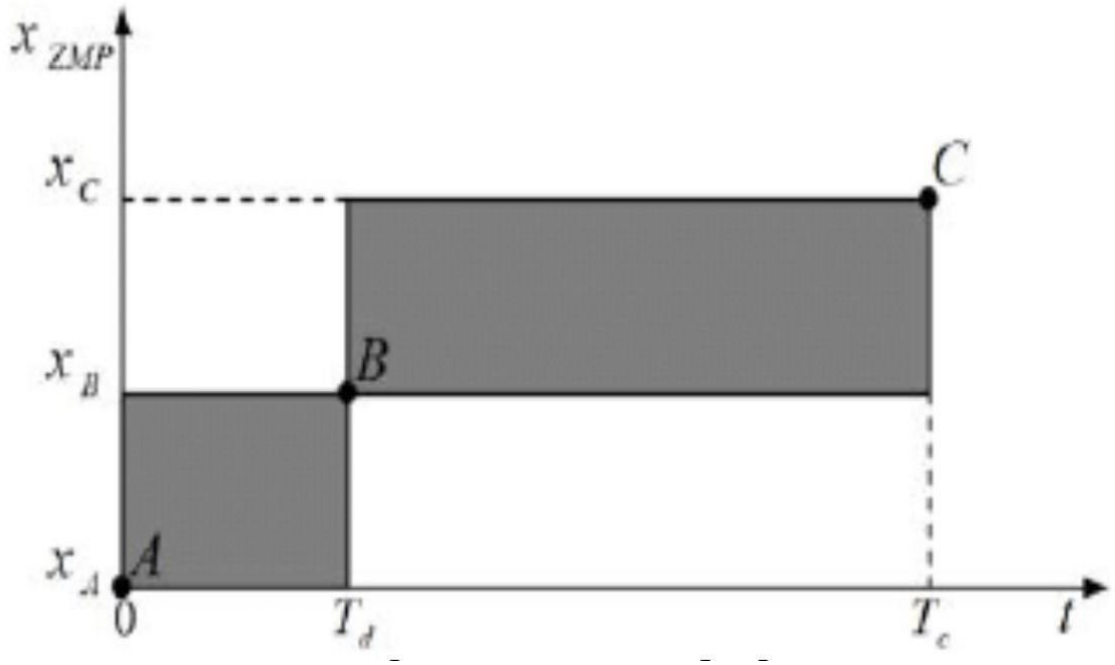

Fig. 16: stable region in a walking step

Using this method, we can also plan trajectories for walking on slope surfaces and stairs (with defining $h_{gs} = h_{ge}$ as stair height).

16.1.2 OPTIMIZATION OF TRAJECTORY

As shown, the position of the highest point of the swing foot (L_{ao}, H_{ao}) does not have any effect on the stability of biped in the sagittal plane. So upon computation of x_{sd} and x_{ed} that guarantee the dynamic stability of robot during the walking, the robot can easily avoid obstacles with different heights. How- ever, for walking on flat ground without any obstacles, the parameters L_{ao} and H_{ao}

can be computed such that the robot has maximum stability or minimum consumed energy during the walking [130]. Since L_{ao} and H_{ao} only affect on trajectory of swing leg ankle joint, the optimization of these two parameters during the walking should be done only in single support phase.

16.1.3 Optimization for maximum stability

Our purpose is to obtain the interested parameters such that the ZMP stays in the stable region and approaches to the midpoint of the stable region as much as possible to have a higher stability margin. For this purpose, the fitness function can be expressed as:

$$f = \int_S [\text{SSP}\, x_{ZMP}(t) - (U_{SR} + L_{SR})]^2 \; dt \quad (52)$$

where USR and LSR are upper and lower stable region

16.1.4 Optimization for minimum consumed energy

Here, our objective is to minimize the total energy of biped during the walking. To this end, the cost function can be expressed as follows:

$$f = \int_R \text{SSP} \sum_{i=1}^{J} |\tau_i \omega_i| \; dt (53)$$

where J is the number of the joints, τ_i is the torque of the joint i, and ω_i is the angular velocity of joint i.

Our trajectory is optimized by using each mentioned fitness functions.

16.1.5 TRAJECTORY PLANNING FOR WALKING WITH DIFFERENT STEP LENGTHS

In previous parts, it is assumed that a robot walks only with the same step length (D_s) in each walking step. However, for walking in environments with different obstacles and ditches, the robot has to change its step length (D'_s) during the walking [131]. Therefore, we define parameters $x'_{ed}, x'_{sd}, x''_{ed}$ and x''_{ed} for hip joint motion for walking cycle that robot has to change its step length to D^s_s as shown in Fig.17.

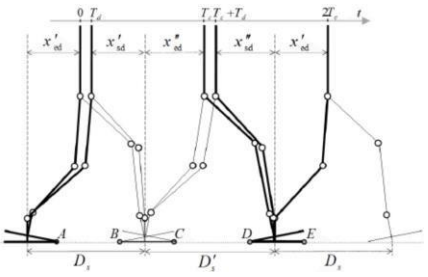

Fig. 17: walking parameters for walking with different step lengths

Trajectory planning for walking with different step lengths that uses for online trajectory planning is similar to method mentioned before but with the following differences:

$$x_{ra}(t) = \{ l_f (1 - \cos\theta_{ra}) + l_a \sin\theta_{ra} \; 0 \le t \le T_d \; L_{ao} t = T_m D_s + D' - l_b (1 - \cos\theta_{ra}) - l_a \sin\theta_{ra} \; T_c \le t \le 2T_c \quad (54)$$

$$x_h(t) = \{ x'_{ed} \; t = 0$$

$$D_s - x'_{sd} t = T_d$$

$$D_s + x''_{ed} t = T_c$$

$$D_s + D'_s - x''_{sd} t = T_c + T_d$$

$$D'_s + D'_s - x'_{ed} t = 2T_c \quad (55)$$

To obtain smooth and periodic trajectories for hip motion, the following constraints must be satisfied:

$$\begin{aligned} \dot{x}_h(t=0) &= \dot{x}_h(t=2T_c) \\ \ddot{x}_h(t=0) &= \ddot{x}_h(t=2T_c) \end{aligned} \quad (56)$$

Stable region in a complete walking cycle is shown in Fig.16. In order to have a stable waking, the ZMP must cross A, B, C, D, and E points. Since walking is a periodic phenomenon, the state of ZMP in A and E points are the same. Therefore, if the ZMP only cross A, B, C, and D points, the biped is able to walk stably. So whenever robot is forced to have different step lengths during walking, the parameters x'_{ed}, x'_{sd}, x_{ed} and x'_{ed} can be determined by the following constraint equations:

$$\begin{aligned} x_{ZMP}|_{t=0} &= x_A \\ x_{ZMP}|_{t=T_d} &= x_B \\ x_{ZMP}|_{t=T_c} &= x_C \\ x_{ZMP}|_{t=T_c+T_d} &= x_D \end{aligned} \quad (57)$$

17 Fuzzy logic controller

Fuzzy logic control (FLC) is well known artificial intelligence, based control technique [132]. It utilizes the prior experience of the functionary about the system to be controlled. The main role of the functionary is to set up decision-based rules by analyzing the system behavior and the linguistic input variables within the framework of the system. The inputs provided to the FLC have to process through three basic stages of fuzzification, decision-making stage, and defuzzification before generating the output [133]. In the fuzzification stage, the input variable is transformed into linguistic variable with the help of predefined membership functions (MFs). The output of the fuzzification stage is then used to generate the fuzzified output according to the rules set defined. Finally, in the defuzzification stage, the fuzzified output is transformed into the required output used for controlling the system. The most interesting fact with respect to the FLC is it does require the exact model of the system during its development. Therefore, The FLC has a extensive scope of applications in the field of machine control for systems that have high uncertainty and nonlinearity in their nature [134]. Consequently, the control strategy of FLC is relatively complex and demands high implementation costs.

Nowadays, fuzzy logic-based controllers are getting more popular in the field of photovoltaic (PV) systems. The PV source generates electricity from solar energy and shows nonlinear I-V and P-V characteristics [135]. Consequently, the output power is nonuniform for all the operating points and has single maximum power point (MPP) for the given atmospheric condition. Thus the PV source which is directly connected to the load is rare to operate at MPP for all the environmental condition. In addition, the dynamic or fast-changing atmospheric condition continuously changes the operating voltage corresponding to MPP and reduces the performance of the PV system. To tackle variable operating voltage due to changing environmental conditions various PV-maximum power point tracking (PV-MPPT) algorithms are given in the literature for the last 20 years [136]. Among the given solutions most classical and frequently used algorithm is perturb and observe (PO) algorithm. The given algorithm is trouble-free and easy to implement. However, it employs the fixed small change in the PV source operating point. It is more suitable for the slow gradual changing environmental conditions. Thus for the fast-changing environmental conditions, it may not be suitable. Here, algorithms using variable step change in the PV operating point may be useful. Such algorithms should be able to predict present operating location with respect to the peak operating point. When the peak operating point is away it may employ a large step change in the PV source operating point to reach near the MPP. And when it reaches near MPP operating it can employ the small step change in the operating point to track exact MPP. Fuzzy logic-based PV-MPPT is one solution for such variable step MPPT algorithm [137]. This part gives detail of both PO and fuzzy-based MPPT algorithms.

The process of FLC control contains the following:

- Fuzzification of the input signals.
- Decide the best solution by the fuzzy decision making logic (DML) based on the expert knowledge base.
- Defuzzification of the solution to obtain the revised angles.

The complete system of fuzzy motion control includes feedback signals from the measurement of a vision system to construct the entire close-loop system. The inputs of the FLC are e and e¨ which are defined as follows:

$$e(t) = -\frac{\theta_d(t)}{90} \quad (9a)$$

$$e(t) = e(t) - e(t-1) \quad (9b)$$

(58)

In order to define the fuzzification interface (FI), the range of the angle error is divided into five fuzzy sets: negative big (NB), negative small (NS), zero (ZE), positive small (PS), and positive big (PB). Then, the membership function for these fuzzy sets is defined in some proper scales. DML utilizes the membership state generated by the FI. Each discourse is divided into five subsets so that the input will construct 25 fuzzy rules. Next, the singleton output u is defined in the defuzzification.

Finally, a suitable turning angle is obtained through the equation that follows:

$$\varphi(t) = C \cdot u(t) \quad (59)$$

where C=90 and is the FLC output that is used in the LPI system to generate the corresponding walking motion. When ——¿M (M is the maximum revisable angle), the robot will stop walking and then turn in the proper direction toward the target.

Fuzzy Logic Controller (FLC) Pseudo Code

```
START
1. Define fuzzy variables (inputs, outputs) and membership functions.
2. Create a rule base (if-then rules) for decision-making.
3. Input current system states (e.g., error, change in error).
4. For each rule:
    a. Evaluate the degree of truth for the rule's conditions.
    b. Combine results using fuzzy operators (AND, OR).
5. Apply defuzzification to generate a crisp output.
6. Update system using the output.
7. Repeat until system stabilizes.
END
```

18 going over an obstacle

18.0.1 Controller objective

We want out controller to be robust enough to track all joint trajectories with full efficacy throughout the trajectory to ascend staircase while ensuring the stability.Along with joint space trajectory tracking, impedance modulation using fuzzy logic controller is proposed to ensure more stable staircase walk. ZMP criteria is considered to ensure the stability of the bipedal robot.

Now lets define the joint space trajectory tracking error vectors $e_1(t)$ and $e_2(t)$ for joint space angular displacement errors and joint space angular velocity error respectively as following:

$$e_1(t) = x_d(t) - x_1(t) 13 e_2(t) = \dot{x}_d(t) - x_2(t) (60)$$

we can have dynamical equations for error vectors as following:

$$\dot{e}_1(t) = e_2(t) 15 \dot{e}_2(t) = -Ae_2(t) + B - I_{in}u(t) - I_{in}Fcon(t) - \Delta d(t) + \ddot{x}_d(t) + A\dot{x}_d(t), (61)$$

where x_d denotes 6×1 desired joint angles vector obtained as output from unsupervised inverse kinematics algorithm. $\dot{x}_d$ and $\ddot{x}_d$ are 6×1 desired joint velocity and joint acceleration vector obtained using finite difference method.

18.0.2 Stability model (ZMP formulation)

One of the most significant tool to test the dynamic stability particularly for bipedal robot is Zero Moment Point. If ZMP lies within the boundary of convex hull formed by various ground contact points, then robot will remain stable and will not fall. x_{ZMP} was formulated in while planning the trajectory using Linear Inverted Pendulum Model (LIPM) based on which we got our hip trajectory [139]. While executing our controllers in presence of external disturbances with our dynamic model, actual ZMP must be within the convex hull to ensure the stability of our bipedal model which can be formulated by following equations in case of staircase:

$$x_{\text{actual ZMP}} = \frac{\sum_{i=1}^{n} m_i x_i(\ddot{z}_i + g) - \sum_{i=1}^{n} m_i \ddot{x}_i z_i}{\sum_{i=1}^{n} m_i(\ddot{z}_i + g) - \beta \sum_{i=1}^{n} m_i \ddot{x}_i} (62)$$

where $\ddot{x}_i$ and $\ddot{z}_i$ are evaluated in Cartesian space to get joint space accelerations and β denotes the virtual slope considered for modeling hip motion based on one-mass COG (Center of Gravity) model. Stability margin is modulus of the difference between convex hull boundary and actual ZMP. A higher stability margin is desired throughout the motion to ensure better preparedness of bipedal robot towards external disturbances or some unwanted events.

18.0.3 Fuzzy logic based impedance modulation

As we know while ascending a staircase, impact plays a very significant role. A significant change in impact force can lead the robot to tip over and finally fall. Magnitude of impact force plays a vital role for a stable landing of the bipedal robot. So to overcome this issue and getting the stable foot landing by modulating the impact forces, we have considered Fuzzy Logic controller to obtain the stiffness and damping matrices denoted by k_s and k_d which are 6×6 diagonal matrices as output with time as fuzzy input. There are mainly two factors to consider fuzzy based impedance modulation:

1) Gait phase inaccuracy: It takes into account the fact that each gait phase namely Double Support Phase 1 (0 to t_1), Double Support Phase 2 (t_1 to t_2) and Single Support Phase (t_2 to t_3) can have some distortion in their respective time bounds due to model inaccuracy or due to external disturbances. By taking time as input to our fuzzy logic controller we can relax time bounds for each gait phase. It can have values Zero, DSP1 for Double Support Phase 1, DSP2 for Double Support Phase 2, SSP for Single Support Phase. Trapezoidal membership function are used for input fuzzification.

2) Load shared by contact points: Particularly in case while ascending or descending staircase there is variance in load or we can say burden carried by different contact point. So the impact force exerted by respective contact point should be according to the burden carried by it at a particular time instant. As we have considered 6 contact points using virtual spring damper model with variable stiffness and damping coefficients. So for $6 \times 6 k_d$ and k_s diagonal matrices, we can have 12 variables $k_{d1}, \ldots, k_{d6}$ and $k_{s1}, \ldots, k_{s6}$. These 12 variables are taken as output to the fuzzy logic controller.

Based on different gait phases fuzzy logic controller rules are in following form:

IF time is X THEN k_{dj} is Y and k_{sj} is Z,

where X can be Zero, DSP1, DSP2, SSP, Y and Z can be Low, Medium, High, Zero and j varies from 1 to 6 for each contact poin.

Fuzzy Logic based Impedance Controller Pseudo Code

```
START
1. Define desired stiffness (K), damping (D), and inertia (M) properties.
2. Input desired trajectory (x_d, x_dot_d).
3. Measure current state (x, x_dot).
4. Calculate force:
    a. F = K * (x_d - x) + D * (x_dot_d - x_dot).
5. Apply force to the system.
6. Update system state.
7. Repeat until the desired motion is achieved.
END
```

19 Neural Networked based control

Artificial neural networks have many advantages, especially when the underlying system dynamics are unclear or difficult to model. One disadvantage, however, is that the network training process is data thirsty. This can be mitigated by

using simulation to pre-train the network. Typically, the simulation needs to run many steps for the network to converge to a good result [139]. Here we first developed a core control system based on a simplified dynamic model. Then we used its outputs to train an artificial neural network (ANN) controller using a reinforcement learning algorithm.We chose to use artificial neural networks (ANNs) with reinforcement learning (RL) techniques to stabilize the biped robot because of the powerful new tools that are available and because of previous successes in applying them to stabilize bipedal walking. For bipedal gait control problems, the RL method is be able to handle continuous input and output. The core control system was developed using classical control methods, and is strongly dependent on the exact mechanical parameters of the model. The RL neural network controller was found to be superior because it is robust to model parameters. With the ANN controller, a different person could use the exoskeleton without changing the controller parameters.

The core control system was based on classical control methods and a reduction in the complexity of the system was essential for its success. To reduce the dimensions of the problem we used principal component analysis [140], whereas virtual constraints for a similar reason [141]. We reduced the dimensions and complexity of the control system by developing separate controllers for the swing and stance phases. To simplify the problem, we assumed that the swing leg's reaction forces have little effect on the torso at the walking speed of the (1 m/s), and our subsequent results justify this assumption. Thus, we divided the control task into two parts: The torso stability task for the stance leg and the optimal foot placement task for the swing leg. The core control system stabilized the biped and rejected impacts and persistent disturbances, but it was not robust to changes in the model.

A neural network was developed and trained using reinforcement learning, starting with data from the core controller. Two ANN controllers were developed: A "local" controller and a "global" controller. The local controller was divided into two sub-controllers, one for swing and one for stance. The advantage of this method is that the action space is small for each sub-controller, so the neural net converged faster. But the disadvantage is that we needed to alternately freeze the parameters of one neural net and train the other one so that they can be better coordinated with each other. Although the stance leg controller is minimally dependent on or entirely independent of the swing foot placement controller based on this assumption, the reverse is not true. The foot placement controller takes both the motion of the swing leg and the stance leg into consideration, and the reinforcement learning optimization process also takes this into consideration. Thus, the parameters for the local neural net swing leg foot placement controller were optimized after the local stance controller was tuned such that the neural net learned the effect of the stance controller on the foot placement controller.

The global controller was trained for the system as a whole rather than separating it into swing and stance controllers. We compared the convergence rates and the performances of the local and global controllers and found that the local controller converged faster but did not perform as well as the globally trained controller.

Neural Network (NN) Controller Pseudo Code

```
START
1. Initialize neural network weights and biases.
2. Train the NN:
    a. Input training data (e.g., state-action pairs).
    b. Perform forward propagation to calculate output.
    c. Calculate loss between predicted and actual values.
    d. Perform backpropagation to adjust weights and biases.
    e. Repeat until convergence.
3. Use the trained NN for control:
    a. Input current system state to the NN.
    b. Compute control output from the NN.
    c. Update system using the output.
4. Repeat for real-time updates.
END
```

19.0.1 DEVELOPMENT OF A CORE CONTROLLER VIA CLASSICAL CONTROL METHODS

The ”core controller” was developed using classical control methods and then used to create data to initialize the neural network controllers to start the reinforcement learning process.

19.1 Stance Leg Controller

The stance leg controller is intended to stabilize the torso during the single-limb support phase. It is designed such that the torso's pitch angle and rotational velocity are small, and its vertical acceleration and velocity are negligible. This greatly simplifies swing foot placement control by representing the single support stance limb as a double inverted pendulum. For the stance leg controller development, we assume:

1. The knee joint angle and its angular velocity are small, so the entire leg can be treated as a single link.
2. The foot/ground friction is large enough so that there is no slipping.
3. The reaction force resulting from the motion of the swing leg is small.
4. The control torque in the ankle is small enough to not cause the foot to rotate about the toe.

The equation of motion of the system can be derived from Lagrange's equation:

$$\frac{d}{dt}\frac{\partial(T-V)}{\partial \dot{q}} - \frac{\partial(T-V)}{\partial q} = Q_1 (63)$$

In matrix form:

$D\,q^{\cdot\cdot} + Cq^{\cdot} + G = Q2$ (64)

$q = [q_1, q_2, q_3, q_4]$ is the set of joint angles representing hip-x (Abduction/Adduction), hip-y (Flexion/Extension), ankle-x (Dorsiflexion/Plantarflexion) and ankle-y (Eversion/Inversion). Hip-z (Lateral/Medial) is locked during stance control. $Q = [Q_1, Q_2, Q_3, Q_4]$ is the control torque that acts on joints $1, 2, 3,$ and 4 . T and V are the kinetic and potential energies of the system. The goal is to find the control torque Q to drive the torso angle P and its velocity $P^{\cdot}$ to zero. The torso angle P is defined as the angle between the torso z-axis and the global z-

axis. Assume a unit vector in the z-direction, $k = [0, 0, 1]^T$, is attached to the torso. Then the vector expression transformed into the global frame is:

$$k'(q) = R_{ankle_x} R_{ankley} R_{hip_x} R_{hip_y} k \quad (65)$$

Where R_{ankle_x}, R_{ankle_y}, R_{hip_x}, R_{hip_y} are rotational transformation matrices associated with each joint. The subscript x and subscript y designate rotation about the x or y axis, respectively. Then, the angle between the k' and world z-axis can be calculated as:

$$P(q) = \text{acos}([0, 0, 1] \cdot k'(q)) \quad (66)$$

By taking derivatives of P, we can calculate the velocity and acceleration of the angle:

$$\dot{P} = \frac{\partial P(q)}{\partial q} \dot{q} \quad (67)$$

$$\ddot{P} = \frac{\partial^2 P(q)}{\partial q^2} \dot{q} + \frac{\partial P(q)}{\partial q} \ddot{q} \quad (68)$$

A constraint equation is imposed on the system so that it will drive the torso angle P to zero:

$$\ddot{P} + k_p P + K_v \dot{P} = 0 \quad (69)$$

Q_1 and Q_2 are control torques on the ankle. These are set to 0 because the ankle is assumed to be an un-actuated joint. One constraint equation is not enough, and there are infinite solutions. Thus, we instead consider the x and y components of vector k'. To minimize the angular rotations of the torso, we desire the x and y components of k' to go to 0 at the same time. But in this way, we must assume that the torso angle P is smaller than $\pi/2$. Otherwise, when we decrease the x and y components of the vector, the angle P will increase instead of decrease. The case where angle P is larger than $\pi/2$ does not happen during normal walking. The velocity and acceleration of the x and y components of k' can be calculated in the same way:

$$X = [1,0,0] \cdot k'(q)$$

$$\dot{X}(q, \dot{q}) = \frac{\partial X(q)}{\partial q} \dot{q}$$

$$\ddot{X}(q, \dot{q}, \ddot{q}) = \frac{\partial^2 X(q)}{\partial q^2} \dot{q} + \frac{\partial X(q)}{\partial q} \ddot{q} \quad (70)$$

$$Y = [0,1,0] \cdot k'(q)$$

$$\dot{Y}(q, \dot{q}) = \frac{\partial Y(q)}{\partial q} \dot{q}$$

$$\ddot{Y}(q, \dot{q}, \ddot{q}) = \frac{\partial^2 Y(q)}{\partial q^2} \dot{q} + \frac{\partial Y(q)}{\partial q} \ddot{q} \quad (71)$$

We design the acceleration so that X and Y are stable and converge to 0 so that the torso remains vertical. Thus, negative position and velocity feedback is used:

$$\ddot{X}_d = -k_{px} X - K_{vx} \dot{X} \quad (72)$$

$$\ddot{Y}_d = -k_{py} Y - K_{vy} \dot{Y} \quad (73)$$

where K_p and K_v are feedback gains for position and velocity. By choosing different gain values, we can manipulate the behavior of these second-order systems. Desired response time and overshoot values can be achieved by using the pole placement method. A large damping value is preferred because oscillation of the torso will decrease the stability of the foot placement controller.

In summary, the following set of equations can be solved for the joint accelerations.

D_u, C_u, G_u are the inertia, centrifugal-Coriolis, and gravity terms for the unactuated joints. Then the control torques can be solved using Eq. 64. Next, a constraint on the ground reaction force is imposed so that there is no foot/ground slip, and the foot stays on the ground for the entire stance phase. The constraints are expressed as:

$$\begin{matrix} F_x^2 + F_y^2 < \mu^2 F_z^2 \\ F_z > 0 \end{matrix} \quad (75)$$

A solution that satisfies these constraint was found using optimization. The variables that need to be optimized are control gains k_p and k_v in Eq. 72-73 for X and Y. The time domain solution can be expressed in terms of a matrix exponent.

$$\bar{X} = \begin{matrix} X \\ \dot{X} \end{matrix} = e^{A_X t} \bar{X}_0 \quad (76)$$

$$A_X = \begin{matrix} 0 & 1 \\ -k_{px} & -k_{vx} \end{matrix} \quad (77)$$

$\bar{Y}$ equations are similar to Eq. 76 for $\bar{X}$. The objective function J is the integral of the square sum of the future error:

$$J = \int_0^T (e^{A_X t} -)^T e^{A_X t} \bar{X}_0 + (e^{A_Y t} -)^T e^{A_Y t} \bar{Y}_0 dt \quad (78)$$

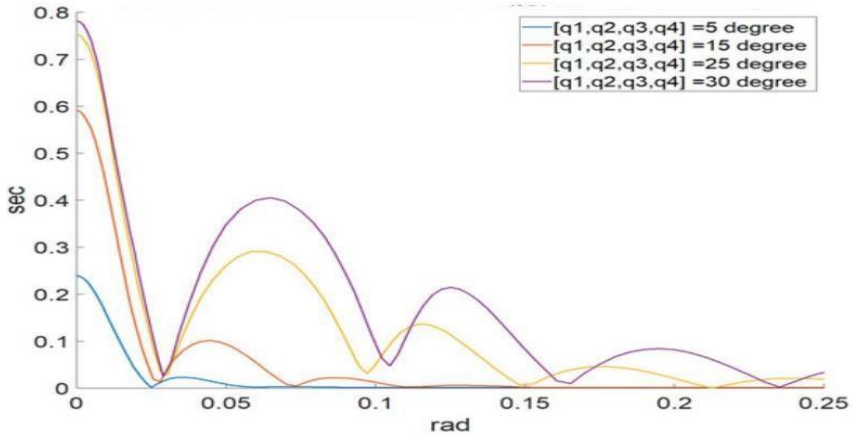

Fig. 18: the time response of the torso angle for different initial conditions

This optimization problem needs to be solved at every time step. It takes approximately 0.02 – 0.05 s using the MATLAB optimization toolbox function "fmincon ()". This was found to consume much computational power and reduce the control frequency. So instead of performing this optimization, the difference between the generated joint acceleration and zero torque joint acceleration were used to limit the torque output of the controller. In a loop, if the constraint is not met, then the control output is reduced:

$$\ddot{q} \leftarrow \ddot{q} + \lambda(\ddot{q} - \ddot{q}_0), \quad 0 < \lambda < 1 \quad (79)$$

$\ddot{q}$ is the zero-torque joint acceleration, and λ is the decay factor. Figure. 18 shows the torso angle control in single limb support phase.

19.2 Foot Placement Controller

After the torso angle is stabilized by the stance leg controller, if the torso angular velocity is small, then the motion of the biped during the swing can be approximated by a single 3-D inverted pendulum. A 3D Linear Inverted Pendulum Model was developed. If the mass center motion plane is parallel with the ground, then the pendulum's time-invariant orbital energy can be calculated by integrating the equations of motion along the x and y axis with respect to time [143].

$$E_y = \frac{1}{2}\dot{y}^2 - \frac{g}{2z}y^2$$
$$E_x = \frac{1}{2}\dot{x}^2 - \frac{g}{2z}x^2 (80)$$

The phase plane of the trajectory shows that if the orbital energy is greater than 0 , then the trajectory will cross the zero position, and if the orbital energy is less than 0 , the trajectory will not. If the orbital energy is equal to 0 , then the trajectory will come to rest at a saddle point, also known as capture point [144]. So, for walking of a 3-D biped in the y direction, for example, then E_y must be greater than 0 , and for stability in lateral motion, it is preferred that E_x is less than 0 so that the COM can oscillate between the two feet. From Eq.80, the initial position can be calculated for the foot placement.

$$y = r\left(\frac{\dot{y}^2}{2} - E_y\right)\frac{2z}{g}$$
$$x = q\left(\frac{\dot{x}^2}{2} - E_x\right)\frac{2z}{g} (81)$$

The calculated initial conditions for foot placement from this model can only generate several steps of stable walking before falling down because of the disturbances from the torso. The model has several restrictions: First, the mass center must move in a plane that is parallel to the ground. Second, the mass center is concentrated around the tip of the pendulum. Third, there is no control input. In practice, it is difficult to satisfy all these constraints[145] bound the nonlinear term resulting from the COM vertical motion to a region where the linear controller still generates a dynamically feasible solution. We added modified terms and coefficients to compensate the difference in modeling:

$$x = \pm d + \mathrm{clip}(x, 1, -1) \cdot \sqrt{\frac{\dot{x}^2}{2} - E_x} \cdot \frac{2z}{g} \cdot (1 + 0.1|P|)$$

$$y = \mathrm{clip}(y, 1, -1) \cdot \sqrt{\frac{\dot{y}^2}{2} - E_y} \cdot \frac{2z}{g} \cdot (1 + 0.1|P|) (82)$$

Where d is the horizontal distance between the hip joint and the center of mass. The clip coefficient regulates the sensitivity of the velocity influence. The coefficient $1 + 0.1|P|$ compensates for the influence from the torso angle. Initial conditions for the center of mass are transformed into foot placement by calculating the forward kinematics. The swing leg trajectory can be generated by inverse kinematics relative to the mass center. In our approach, the inverse kinematics for x and y are relative to the mass center, and z is relative to the global reference frame. In this way, the z component of the foot displacement will

not be affected by sudden movements of the mass center and, thus, the ground contact is more controllable. The position of the foot X_f can be expressed as:

$$\begin{bmatrix} 0 \\ 0 \\ z_c \end{bmatrix} + R \cdot (X_{c-h} + F(q)) = X_f \quad (83)$$

Where z_c is the z component of the mass center in global coordinates and R is the rotation matrix from global coordinates to body-fixed coordinates. $F(q)$ is a vector of the Cartesian coordinates of the foot relative to the hip expressed as a function of the joint angles q.X_f is the foot coordinates partially relative to COM in global coordinates. X_{c-h} is a vector from the mass center to the hip joint in bodyfixed coordinates:

$$X_{c-h} = X_{GC-hip} - X_{GC-COM} \quad (84)$$

X_{GC-hip} is a vector from the torso's geometric center to the hip joint in the body-fixed coordinates, and X_{GC-COM} is a vector from torso's geometric center to the mass center in body-fixed frame. X_{GC-COM} is a function of the joint angles q. Taking the time derivative of both sides of Eq. 83:

$$\begin{bmatrix} 0 \\ 0 \\ \dot{z}_c \end{bmatrix} + \dot{R} \cdot (X_{c-h} + F) + R \cdot (\dot{X}_{c-h} + \dot{F}) = \dot{X}_f \quad (85)$$

$\dot{R}$ is a function of Euler angles (α, β, γ) and their time derivatives.

$$\dot{X}_{c-h} = \dot{X}_{GC-hip} - \dot{X}_{GC-COM} = -\dot{X}_{GC-COM} = -J_{GC-COM}\dot{q} \quad (86)$$

J_{GC-COM} is the system Jacobean:

$$J_{GC-COM} = \frac{\partial X_{GC-COM}}{\partial q} \quad (87)$$

J_{GC-COM} can be expressed as $[J_{GC-COM-l}, J_{GC-COM-r}]$, and $\dot{q}$ can be written as $\begin{bmatrix} \dot{q}_l \\ \dot{q}_r \end{bmatrix}$. The subscript" l" and" r" represent the left and right leg, respectively. Then:

$$\dot{X}_{c-h} = -J_{GC-COM}\dot{q} = -J_{GC-COM-l}\dot{q}_l - J_{GC-COM-r}\dot{q}_r \quad (88)$$

And

$$\dot{F} = J_k\dot{q}_l \text{ or } \dot{F} = J_k\dot{q}_r \quad (89)$$

Depending on which leg is the swing leg. J_k is the foot position Jacobian for leg kinematics. For example, if the left leg is the swing leg, then substituting equations

$$R \cdot (X_{GC-\text{hip}-l} - X_{GC-\text{COM}} + F_l) + \dot{R} \cdot (-J_{GC-\text{COM}} - r\dot{q}_r + (J_k - J_{GC-\text{COM}-l})\dot{q}_l) \quad (90)$$

$$\begin{bmatrix} \square \\ 0 \end{bmatrix}$$

And using inverse kinematics, the left leg joint velocity can be calculated:

$$\dot{q}_l = (RJ_k - RJ_{GC-COM-l})^{-1} \cdot (\dot{X}_{GC-hip-l} - \dot{X}_{GC-COM} + \dot{F}_l + RJ_{GC-COM-r}\dot{q}_r) \quad (91)$$

This equation is then integrated to find the desired joint angles.

19.3 DEVELOPING AN ARTIFICIAL NEURAL NETWORK CONTROLLER AND TRAINING IT WITH REINFORCEMENT LEARNING

the core controller derived in the previous part stabilized the biped for which it was designed. However, it was found to be brittle. It became unstable with even a small change in biped parameters. Thus, to create a more robust controller, an ANN was designed and trained using reinforcement learning starting with data from the core controller.

The neural networks for enhancing the core controller need to be able to work with continuous inputs and outputs. There are many reinforcement learning (RL) techniques that can be used to optimize a neural net in such a case. Many are gradient-based and require large samples of trajectories. The training algorithm used in this work is called Proximal Policy Optimization (PPO) p[146]. It is a popular off-policy gradientbased optimization method. PPO is the default learning algorithm for Open-AI because it is efficient compared to many on-policy stochastic policy gradient methods, and it is straightforward to implement compared to its full version: Trust Region Policy Optimization (TRPO) [147]. For a policy gradient method, an agent interacts with its environment, observes its state s, and then outputs action a according to policy π_θ, and then the agent will move to the next state according to action a and so on. A trajectory $\tau\,(s_1, a_1, s_2,\ a_2, s_3, a_3 \ldots)$ of state and action can be recorded. The probability for τ is

$$\mathrm{p}(\tau) = p\,(s_1)\,p_\theta\,(a_1 \mid s_1)\,p\,(s_2 \mid a_1, s_1)\,p_\theta\,(a_2 \mid s_2)\,p\,(s_3 \mid a_2, s_2)\ldots \quad (92)$$

p_θ represents the possibility for the agent to output a certain action given the state. This possibility is controlled by the policy parameter θ. The goal is to train this agent so that it will have a high possibility to output the action that can lead to a larger reward. The policy gradient method is an on-policy method, meaning that the data used to calculate the gradient must be gathered by the current policy. Otherwise, a different policy will give a different distribution, and the sampled gradient will not approximate the true gradient. Thus, every time the policy parameter is updated, all the data collected previously will be outdated and cannot be used in the future. This results in the policy gradient method spending most of its computational time collecting data rather than training.

PPO uses importance sampling to mitigate this problem [148]. Importance sampling allows one to calculate the expectation of a distribution p from a different distribution q. However, this expression becomes more difficult to approximate when the difference between the two distributions is large. So, in the PPO algorithm, a KL divergence is added to the objective function to measure the difference between distributions [149]. This KL divergence is used to penalize large differences between distributions. By using this approach, the

data can be used to update the parameters several times. PPO can also use a modified surrogate objective $L^{CLIP}(\theta)$ to limit the step size during a trust-region optimization update.

As mentioned previously, the goal is to develop a neural network trained using reinforcement learning initialized with data from the core control system. But the structure of the core control system separates the full biped action control into two parts: swing leg control and stance leg control. On the one hand, this reduces the difficulty of the design of each controller. On the other hand, it increases the training difficulty for the reinforcement learning algorithm. Two different methods of implementing the learning algorithm are used in this work and the results are compared.

19.4 Local Reinforcement Learning

In the first method, called the ”local” method, there are separate neural networks for each action control task: A foot placement controller neural network and a stance leg controller neural network. Because the control task was divided into two parts, the optimization task was eased as compared to trying to optimize a single more complex control system for the entire interconnected dynamic system. Thus, we could find sufficient RL policies for each of these two control problems with fewer iterations.

For the purpose of control, the gait cycle was divided into swing, double-support, stance, and toe-off (we treat the toe-off as a gait phase). Except for the double-support phase, the timing of the phase switching was controlled by feedback from contact sensors at the foot and the output of the controllers. The duration of the double support phase was linearly related to the torso velocity. Simulated experiments showed that this intuitively derived linear relationship was sufficient to represent the system. However, in future work, it could also be optimized using a neural net.

Toe-off can be achieved in two ways. One is to generate a trajectory for the foot using inverse kinematics for the target joint angle controller to lift the foot. The other method is to disable the knee target joint angle controller and apply a direct reflexive torque to flex the knee. This is necessary because the disturbance caused by toe-off often causes the swing foot to hit the ground when the foot does not follow the designed trajectory, and a target angle controller will make the knee joint stiff. It is inefficient and may destabilize the system. Flexing the knee allows the swing foot to move forward freely without the knee pushing the foot against the ground. In normal walking, when there are no disturbances, the trajectory following method is used. But, when a premature foot contact is detected, the controller switches to the reflexive method.

The simulations proceeded as follows. The foot placement controller neural network was active once every footstep. The input vector included torso linear and angular velocities in the x and y directions, torso angle, torso height, and which leg (swing leg) it should control at that instant. The outputs were the target hip joint angle in the x-axis, which controlled how far the biped should

step, and the duration of the step, which determined how fast the joint should rotate, and the timing of the knee extension. The stance leg controller neural network was active at 50hz during stance. The input vector to the controller was the same as the foot placement controller plus additional stance leg joint angle and joint velocity information. The output was the hip joint velocity of the stance leg. The reason to use target joint velocity control is that it can improve policy performance and learning speed [150]. The environmental reward function used for the foot placement controller neural network training was simply the forward travel distance of the biped, and the reward for the stance leg control controller neural network was the norm of the torso angle and angular velocity vector. Each episode ended when the mass center height was below the threshold or the travel distance reached a preset maximum value.

After the foot placement controller neural network was optimized, its parameters were frozen, and we then optimized the stance leg controller neural network. The optimization of the two neural networks changed the dynamics of both the stance leg controller and the foot placement controller. Thus, it was found useful to iterate this process so the neural network in each controller could adapt to changes and synergize better with the other. We have found that this training loop was prone to converge because, while the foot placement controller is strongly influenced by the stance controller, the stance leg controller depends little on the foot placement controller. Also, if the simulation displayed some unrealistic behavior due to the numerical solver in Gazebo, then the entire trajectory collected in that episode was not used in training.

19.5 Global Reinforcement Learning

In the second method, called the ”global” method, only one neural network is trained, and it controlled both the swing leg and the stance leg. The action space was much larger compared to the local method. This neural network was run at 50hz. The state input was a series of state vectors consisting of torso position and velocity, mass center relative position and velocity, joint angle and velocity, left and right contact sensors, and the output from the core controller. The state also included a target coordinate and the target speed. The input state series allowed the neural net to perceive some events that are hard for a single timestamp state input to represent, such as the ground contact and foot slipping. The length of the input series was set to 5 so that a total of 0.1 s of motion was recorded in the state series in a 50hz control loop. This is similar to a human’s 0.15 s reaction time for a touch stimulus. The environmental reward was the error norm of current torso position to the target coordinate and current speed to target speed.

The output of the neural network was stance leg joint velocity (hip-x, hip-y, knee) and x, y and z coordinates of the predicted foot placement for the swing leg. The final output of the control system was the summation of the output from the neural net and the core controller. At the beginning of the training, the neural net was initialized to output zero means and small variances so that

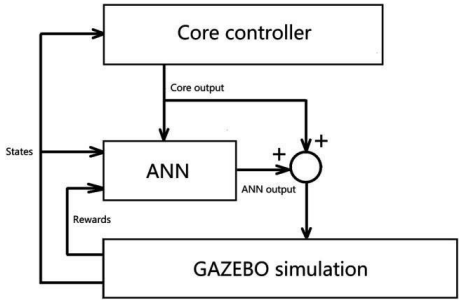

Fig. 19: The relationship between the core controller and the ANN

the total output was similar to the output of the core controller. Thus, the agent started its exploration near an optimal point. After the first parameter update of the neural net, the agent departed from the initialized parameters, and the stable gait provided by the core controller was no longer imposed. After some iterations, the agent optimized the policy and generated a stable gait. The RL-trained neural network controllers were much more robust to changes in the model's mechanical parameters than the core controller.

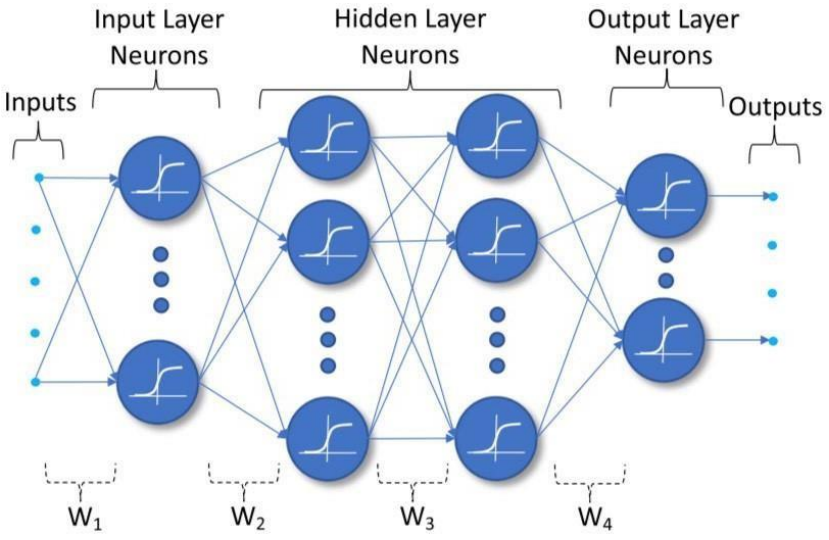

Fig. 20: Neural Networks Architecture

the core controller for both methods and the policy in the local reinforcement learning method use only the current state of the biped. The global reinforcement learning method policy uses the current and a series of past states of the biped as input. In the global reinforcement learning method, the neural net

output modifies both foot placement control and stance leg control. While in the local reinforcement learning method, one neural net is trained to modify the foot placement control output, and another is trained to modify the stance leg control output. The switching between left-swing-right-stance and right-swing-left-stance is triggered by signals from the contact sensors. All the measured signals are sampled from the simulated environment. The foot placement controller converged relatively quickly in Method 1, the local reinforcement learning method. It needed about 15,000 walking steps and 150 iterations for a good policy to emerge, while other tasks have been reported to need 10^5 to 10^7 steps.the global reinforcement learning method, needed longer trajectories in each iteration for training, and the overall convergent speed was slower than the local reinforcement learning method. The simulation ran in real time, and only one biped model was simulated.

20 Chapter 4: The results and conclusions

20.1 RESULTS

The robustness of the control system was tested by adding a gradually increasing short-duration (impact) force on the center of the torso during walking until the biped fell [151]. There was a time delay between two consecutive forces to allow the biped model to return to its normal walking before the next impact. The duration of the impact force was 0.1 s. The test was performed with force applied to the torso in different directions and repeated six times for each direction. The direction of the impact force was varied from $-\pi/2$rad (rearward) to $\pi/2$rad (forward) for every 0.1rad. The results for the model are shown in Figure 21.

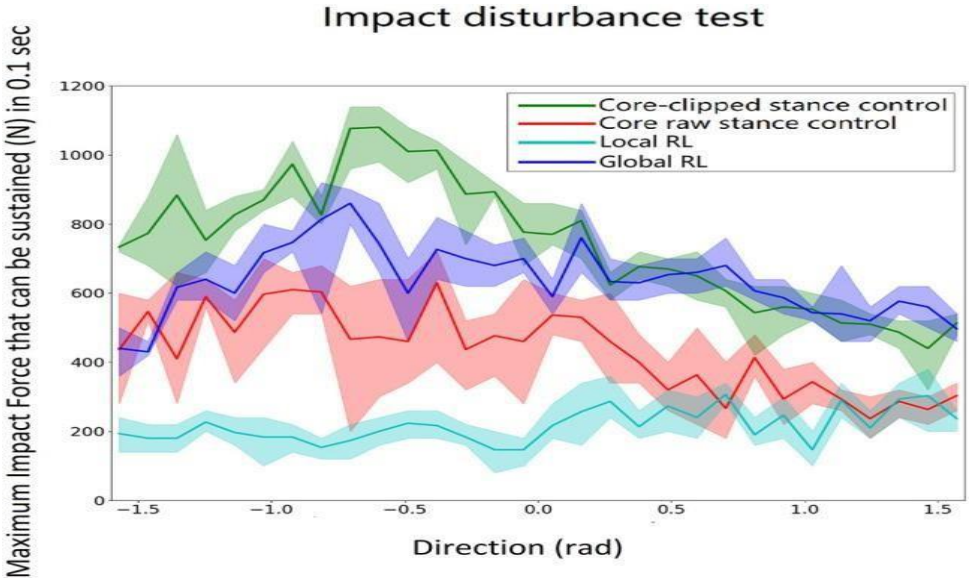

Fig. 21: Impact test

As can be seen in Figure 21, all four controllers (core-clipped, raw-core, global reinforcement learning method, local reinforcement learning method) generated stable gaits. However, the stable gaits generated by the neural network controllers were less robust to impact disturbances compared to the core-clipped controller.

the performance of the system was tested when a constant force was added for a long period of time. This tested the system's robustness against persistent disturbances such as wind. The force was added on the center of the torso link for 20 s. The global neural net controller can stabilize much larger persistent forces in the forward and lateral directions compared to the rearward direction. The velocity of the biped was decreased by the rearward force. The forward force caused a short time of increasing speed, but the controller adapted to slow the pace. After the force was cleared, the biped took about 5 s to recover to normal walking. The lateral force caused little change of the lateral speed because the neural net was trained to walk forward. So, the lateral error caused by the disturbance force was compensated by the controller. This result is shown in Figure 22.

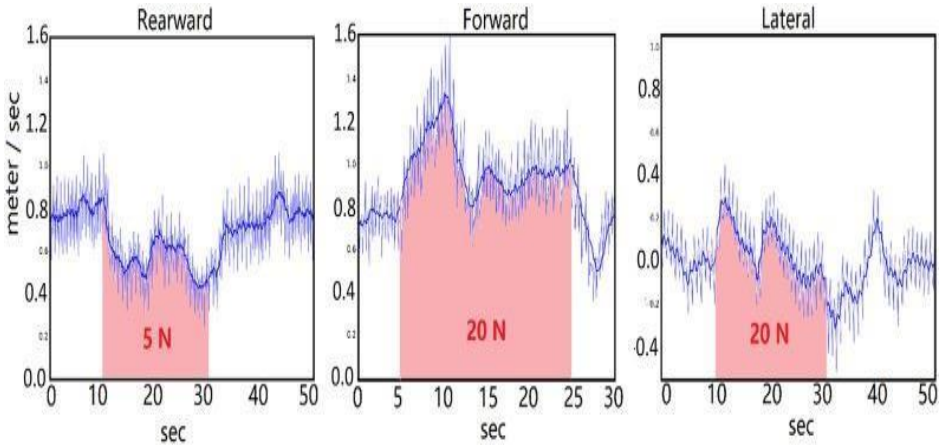

Fig. 22: Persistent disturbance test

The neural network for the PPO policy and critic networks each have three densely connected layers with 400 units per layer. The activation function is Leaky-ReLu. Other parameters are listed in Table 3.

Table 4: neural network parameters.

Parameter	Value
Actor learning rate	1e − 5
Critic learning rate	1e − 4
Critic loss type	L1
Batch size	512
Num of batches	100
Max episode steps	32,000
Gamma	0.995
GAE-lambda	0.95
PPO-epsilon	0.2

The results show that the core-clipped controller, developed using classical control methods, is more robust to external perturbations than the reinforcement learning policy. However, the core-clipped controller is brittle to even small changes in the plant. For example, if the user picks up a bag of groceries,

the core clipped controller will have to be re-tuned. This is unacceptable for an exoskeleton.

Instead of using the core controller alone, we used the core controller's outputs and the states of the biped as the inputs to a neural network and then trained the network using reinforcement learning. The trained neural network was then shown to control stable walking of the biped. We found this method has three advantages. First, because the core controller provided a good initial solution, the network converged relatively rapidly compared to results reported in the literature. Second, the NN controller produced stable walking of the three-dimensional biped that was robust to external perturbations and drastic changes in the biomechanical model. Third, additional goals for the controller were able to be realized because of the flexible choice of different reward functions for the RL training process [153].

The global reinforcement learning policy shows superior performance to the local reinforcement learning policy. In the global reinforcement learning policy, stance and swing are learned in parallel, whereas in the local reinforcement learning policy, stance and swing are learned separately. Thus, the local reinforcement learning policy is the sum of two smaller problems and converges much more rapidly.

This method is attractive for use in biped robot control because it does not require a detailed dynamic model of the particular user. In future work, the joint control signal can be transformed into muscle/motor activation for muscle-first driven hybrid exoskeletons [154].

This method can be used to evaluate the efficacy of additional joint degrees of freedom . Adding degrees of freedom to a biped robot is done with care because of the added weight, complexity, and cost. Most exoskeleton's confine the user to leg movements in the sagittal plane, which has drawbacks in terms of normal joint movements, walking speed, and stability. Additional joints can be easily added to or subtracted from a dynamic model and this method can be used to learn a controller. The performance of the system with the additional joints in terms of speed, stability, and robustness to perturbations and changes to the mechanical model (different users and carrying objects) can be evaluated. Thus, the method described in this thesis can be an important tool for design as well as system control.

overall we first used classical control methods to design a core control system consisting of a stance-leg torso stabilization controller and a swing-leg foot placement controller. The stance leg controller was based on a double pendulum model. The acceleration of the torso angle was designed to mimic a stable second-order system. Then the required torque was calculated by using double pendulum dynamics. The output was then modified to further stabilize the system. After that, the swing leg foot placement prediction controller was designed using the modified orbital energy method. The resulting" core" controller generated stable walking and was robust to impact disturbances. But it cannot stabilize a biped with different biomechanics because the controller was designed on precise model data. Thus, we used reinforcement learning to train a neural network for biped walking using the outputs of the core controller and

the states of the biped as its inputs. Two different ways of training the policy were tested. The local reinforcement learning method used separate neural nets for swing leg control and stance leg control so that each of these neural nets only controls a portion of the action and can be trained with less computational cost. The global reinforcement learning method used just one neural net for the entire action control of the system. Results showed that the local reinforcement learning method indeed trained faster than the global one, but it is less robust than the global reinforcement learning method.

21 Error Signals

Control theory is a field of control engineering and applied mathematics that deals with the control of dynamical systems in engineered processes and machines. The objective is to develop a model or algorithm governing the application of system inputs to drive the system to a desired state, while minimizing any delay, overshoot, or steady-state error and ensuring a level of control stability; often with the aim to achieve a degree of optimality. To do this, a controller with the requisite corrective behavior is required. This controller monitors the controlled process variable (PV), and compares it with the reference or set point (SP). The difference between actual and desired value of the process variable, called the error signal, or SP-PV error, is applied as feedback to generate a control action to bring the controlled process variable to the same value as the set point. Other aspects which are also studied are controllability and observability. Control theory is used in control system engineering to design automation that have revolutionized manufacturing, aircraft, communications and other industries, and created new fields such as robotics [155].

here we presented the Error signals of our 3D biped robot in the first scenario (walking on the flat surface) being controlled via PID, FLC and Neural Network controllers and depict which controlling method is more accurate one. first scenario is when the biped robot is walking on a flat surfaced and PID control is used in this stage. the results are shown here: As you can see it does take

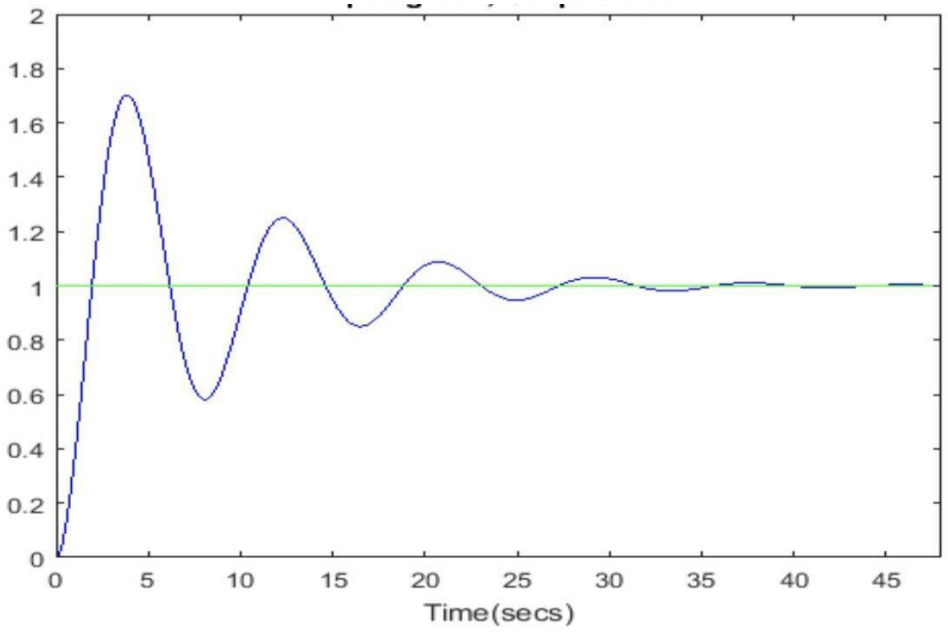

Fig. 23: PID control error signal

some time for the error signal to become stable. As you can see FLC controller is more stable than PID controller and the robot gets stable while walking sooner with FLC controller compared PID controller. As the figures depict NN controller the most stable and fattest controller compare to FLC and PID and more efficient one compare to others. So we conclude that NN controller is way more accurate compare to the 2 other ones.

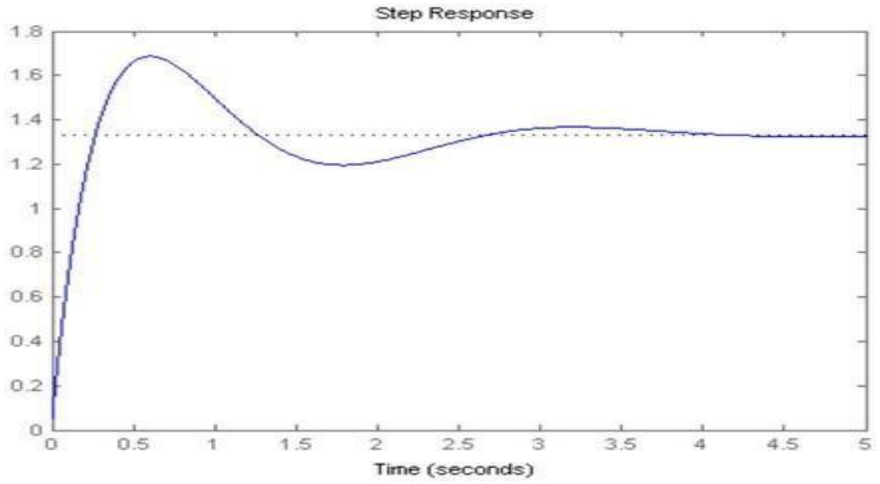

Fig. 24: FLC control error signal

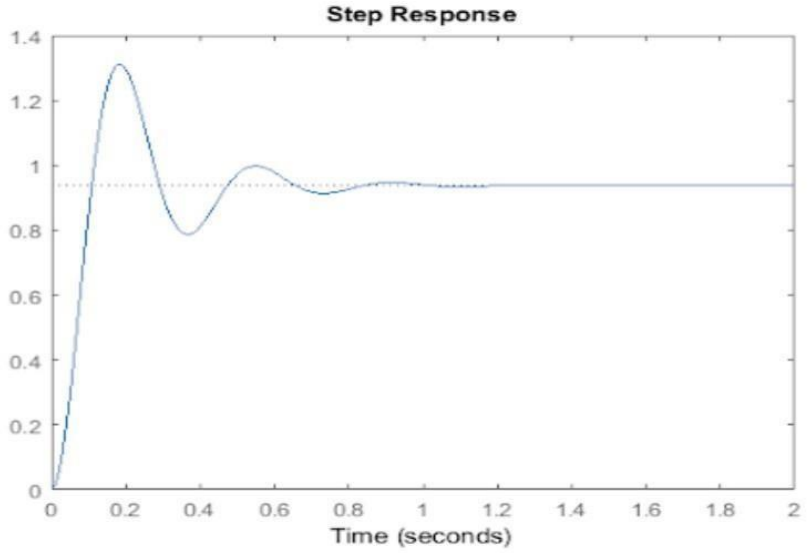

Fig. 25: neural network controller error signal

22 Conclusion

Comparing real and desired trajectories after applying each controller on the biped walking robot in different scenarios In this scenario we are applying NN controller to the biped walking robot on the flat surface, the trajectory is as shown below:

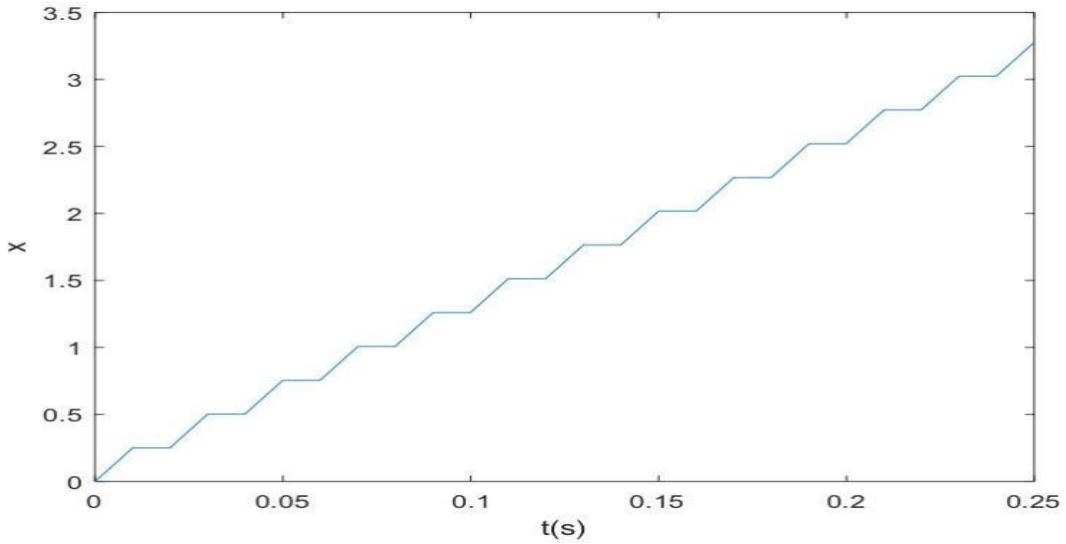

Fig. 26: biped robot trajectory after NN controller applied on the flat surface

the second scenario is after applying NN controller to the biped walking robot while facing an obstacle, the trajectory is as shown below:

the third scenario is after applying NN controller to the biped walking robot while passing over a ditch, the trajectory is as shown below:

It can be seen that bipedal robot has most stable trajectory while walking on the flat surface as the real trajectory is completely like the desired one. the second place is while facing an obstacle and finally it's when over passing a ditch.

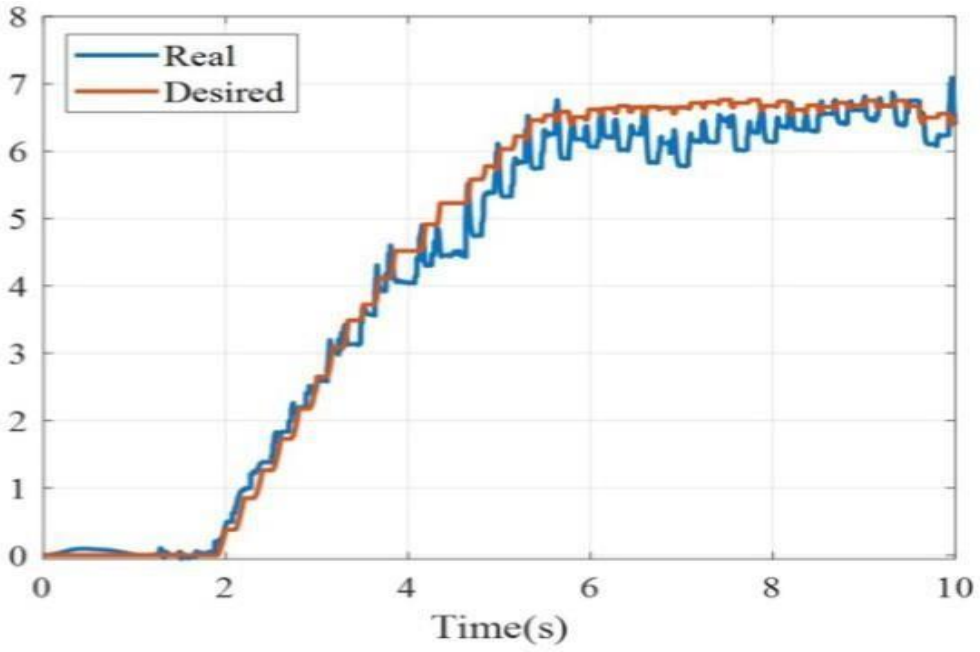

Fig. 27: biped robot trajectory after NN controller applied while facing an obstacle

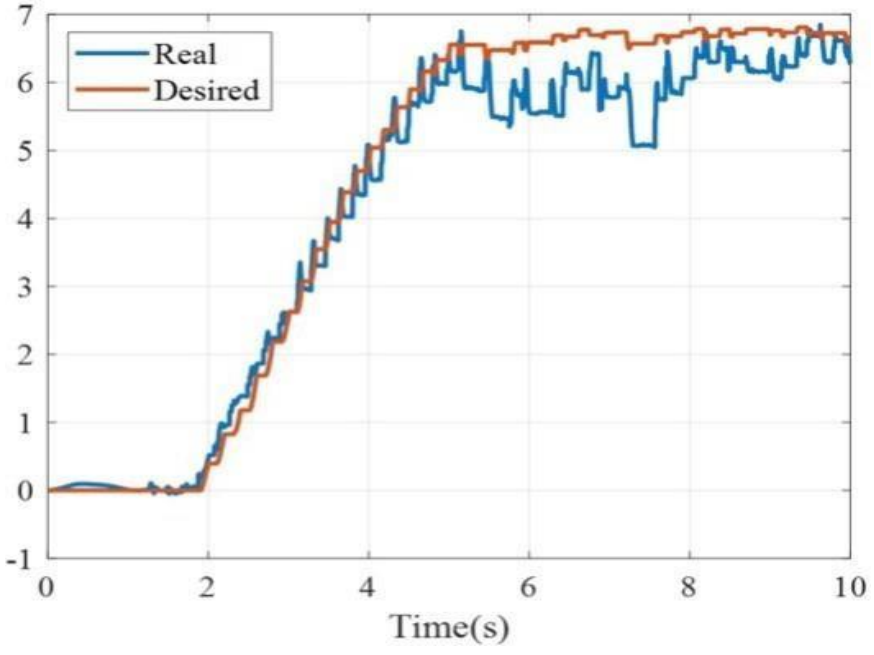

Fig. 28: biped robot trajectory after NN controller applied while passing over a ditch

In this work we only could demonstrate all the PID results due to the limited time. Here are the reasons why we chose the PID controller to fully cover over the other controlling approaches.

1. Simplicity of Design and Implementation

The design of a PID controller involves straightforward tuning of three parameters (Proportional, Integral, and Derivative gains). This simplicity makes it computationally light and easy to implement, particularly in systems with limited computational power or where cost-efficiency is

critical.

Neural Networks (NN) and Model Predictive Control (MPC), while powerful, require significant computational resources for training and prediction, respectively. For example:

- NNs involve iterative learning algorithms and matrix operations, which can be resource-intensive.
- MPC requires solving optimization problems at each control step, which adds a considerable computational burden.

2. Real-Time Responsiveness

Its algorithm is based on simple arithmetic operations (error calculation, summation, and differentiation). This ensures minimal latency, enabling fast real-time adjustments.

- Fuzzy Logic Controllers (FLC) must evaluate multiple fuzzy rules, increasing computational overhead as the number of rules grows.
- NNs may introduce delays due to their forward propagation calculations, especially in larger networks.
- MPC, due to its predictive nature, can struggle with real-time execution when dealing with high-dimensional systems.

3. Low Computational Load

PID calculations scale linearly with the number of control parameters, making it ideal for applications where computational resources are constrained.

- NNs involve nonlinear computations and backpropagation during training, leading to quadratic or cubic complexity depending on the network's size.
- CTC requires a detailed dynamic model of the system and involves matrix inversions, which can be computationally expensive.
- MPC's predictive optimization is computationally intensive, often requiring high-performance hardware for real-time applications.

4. Cost-Efficiency

The simplicity of PID makes it cost-effective to implement in both software and hardware. It does not require specialized processors or extensive training data.

- NNs and MPC often demand expensive hardware (e.g., GPUs or multicore processors) to handle their computations effectively.
- The design and tuning of FLC require domain expertise and iterative testing, adding to development costs.

5. Performance in Structured Environments

PID excels in structured and predictable environments like flat terrain or controlled obstacles. It provides smooth and stable control with minimal effort.

- FLC, NN, and MPC are more suitable for dynamic or unstructured environments but introduce unnecessary complexity and cost when applied to simpler problems.

All in all, The PID controller's low computational cost, ease of implementation, and fast real-time performance make it an ideal choice for problems where resource constraints and simplicity are prioritized. While other controllers offer advanced features, their computational overhead and cost often outweigh the benefits in structured or moderately dynamic scenarios. For the tasks demonstrated in the document—such as walking on flat surfaces or handling simple gait dynamics—PID achieves efficient and effective results without the complexity of alternative approaches.

Overall, this thesis comprehensively investigates the intricate mechanisms and methodologies required to achieve stable, efficient, and adaptive locomotion for bipedal robots in a variety of environments. Addressing one of the most complex challenges in robotics—the emulation of human-like walking—the research delves into the interplay between control strategies, dynamic stability, and the ability to handle real-world uncertainties. By focusing on the design and control of a 3D biped robot, the study systematically evaluates and compares various control approaches, offering a nuanced understanding of their strengths, limitations, and applicability across different terrains and scenarios.

A cornerstone of this work is the application of diverse control techniques, including PID controllers, Fuzzy Logic Controllers (FLC), Neural Networks (NN), Model Predictive Control (MPC), and Computed Torque Control (CTC). Each controller brings unique capabilities to the table: PID controllers provide a simple yet effective mechanism for addressing basic locomotion tasks, although their effectiveness diminishes with increasing uncertainty and complexity. Fuzzy Logic Controllers excel in managing nonlinearities without requiring precise

models, offering robustness in unpredictable scenarios. Neural Networks, with their learning-based adaptability, enable the robot to refine its control policies in real-time, making them suitable for dynamic and evolving environments. Model Predictive Control stands out for its ability to predict and optimize future states, providing an advanced framework for handling multivariable systems and constraints. Lastly, Computed Torque Control demonstrates how precise modeling can lead to improved trajectory tracking and dynamic stability, albeit with the caveat of requiring highly accurate system representations.

This study underscores the importance of stability criteria such as Zero Moment Point (ZMP) analysis, which plays a critical role in ensuring the robot's balance during locomotion. By integrating ZMP with techniques like trajectory interpolation and limit cycle analysis, the study bridges theoretical constructs with practical implementations. This blend of theoretical rigor and experimental validation is further enriched by the use of biologically inspired mechanisms, such as Central Pattern Generators (CPG), which emulate the rhythmic locomotion observed in humans and animals. These approaches not only enhance stability and adaptability but also reduce computational overhead by leveraging natural dynamics.

The exploration of gait generation across varied terrains—including flat surfaces, slopes, staircases, and uneven or unknown environments—reveals the versatility and resilience required for real-world applications. On flat terrain, simpler model-based techniques suffice, but as the complexity of the terrain increases, more sophisticated controllers and adaptive algorithms become necessary. The ability to ascend and descend slopes, navigate obstacles, cross ditches, and maintain stability on uneven ground showcases the breadth of capabilities that a well-designed biped robot can achieve. In particular, the integration of sensory feedback, such as gyroscopes, accelerometers, and force sensors, enables the robot to perceive and respond to environmental changes in real time, further enhancing its autonomy and effectiveness.

The simulation results in the Gazebo environment provide a robust testing ground for validating these control methods and assessing their performance under various conditions. By testing the robot in scenarios such as walking on flat surfaces, crossing ditches, and avoiding obstacles, the study highlights the practical

challenges and solutions for achieving smooth and balanced locomotion. The comparisons between control methods reveal the trade-offs between simplicity, computational efficiency, and adaptability, offering valuable insights for future research and development.

This work not only advances the field of humanoid robotics but also lays a foundation for broader applications in healthcare, service robotics, exploration, and disaster response. The ability to navigate challenging environments with agility and stability has implications for improving the quality of life and extending human capabilities in environments that are hazardous or inaccessible. Furthermore, the integration of biologically inspired and learning-based techniques suggests a future where robots can seamlessly interact with their surroundings, adapting and improving their behavior over time.

In conclusion, this work represents a significant step toward achieving human-like locomotion in bipedal robots. By combining diverse control strategies, stability criteria, and innovative design principles, it demonstrates the potential for creating robots that are not only functionally effective but also computationally efficient and adaptable to real-world conditions. The work invites further exploration into hybrid control methods, enhanced sensory integration, and real-time decision-making algorithms to address the remaining challenges in bipedal locomotion. As robotics continues to evolve, this research serves as a vital contribution to the ongoing quest for machines that move, think, and adapt like humans.

23 Abbreviations

- AI artificial intelligence
- ASD autism spectrum disorder
- BBWR bird biped walking robot
- BCI brain-computer interface
- BLDC brushless DC
- COG center of gravity
- COT cost of transport
- CPG central pattern generator
- DC central drive of legs move
- DG independent balancing drive
- DL drive of left foot
- DR drive of right foot
- DC direct current
- DOFs degrees of freedom
- DSP double support phase
- EHA electro-hydrostatic actuator
- FLC Fuzzy Logic Control
- HBWR human biped walking robots
- ICT information and communications technology
- IMU inertial measurement unit
- LIDAR light detection and ranging
- NN Neural Network
- PID proportional–integral–derivative
- RL Reinforcement Learning
- SBWR synthetic biped walking robots
- SMA shape memory alloys
- SR specific resistance
- SSP single support phase
- VO visual odometry
- ZMP zero moment point

24 References

1 [1] Hobon, M., De-León-Gómez, V., Abba, G., Aoustin, Y. and Chevallereau, C., "Feasible speeds for two optimal periodic walking gaits of a planar biped robot," Robotica 40(2), 1–26 (2021). doi: 10.1017/S0263574721000631.Google Scholar

[2] Vukobratovi´c, M. K., "Contribution to the study of anthropomorphic systems," Kybernetika 8(5), 404–418 (1972).Google Scholar

[3] Vukobratovi´c, M. and Stepanenko, J., "On the stability of anthropomorphic systems," Math. Biosci. 15(1-2), 1–37 (1972). doi: 10.1016/0025-5564(72)90061-2.CrossRefGoogle Scholar

[4] Seo, Y.-J. and Yoon, Y.-S., "Design of a robust dynamic gait of the biped using the concept of dynamic stability margin," Robotica 13(5), 461–468 (1995). doi: 10.1017/S0263574700018294.CrossRefGoogle Scholar

[5] Vukobratovi´c, M. and Borovac, B., "Zero-moment point—thirty five years of its life," Int. J. Hum. Robot. 1(01), 157–173 (2004).10.1142/S0219843604000083CrossRefGoo Scholar

[6] Hobbelen, D. G. E. and Wisse, M. Limit cycle walking In: Humanoid Robotics (M. Hackel ed.), (I-Tech Education and Publishing, Vienna, Austria, 2007) pp. 277–294.Google Scholar

[7] Huang, Q. and Ono, K., "Energy-Efficient Walking for Biped Robot Using Self-Excited Mechanism and Optimal Trajectory Planning," In: Humanoid Robots: New Developments (2007).Google Scholar

[8] Kajita, S., F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, "Biped Walking Pattern Generation by Using Pre- view Control of Zero-Moment Point," IEEE International Conference on Robotics and Automation. IEEE ICRA 2003 (2003) pp. 1620–1626. doi: 10.1109/ROBOT.2003.1241826.CrossRefGoogle Scholar

[9] Vundavilli, P. R. and Pratihar, D. K., "Gait Planning of Biped Robots Using Soft Computing: An Attempt to Incorporate Intelligence," In: Intelligent Autonomous Systems: Foundations and Applications (Pratihar, D. K. and Jain, eds.), L. C. (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010) pp. 57–85. doi: 10.1007/978-3-642-11676-6$_4$.*CrossRefGoogleScholar*

[10] Zheng, Y. F., "A Neural Gait Synthesizer for Autonomous Biped Robots," IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications (1990) pp. 601–608. doi: 10.1109/IROS.1990.262457.CrossRefGoogle Scholar

[11] Yin, Y. and Hosoe, S.," Mixed Logic Dynamical Modeling and on Line Optimal Control of Biped Robot," 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (2006) pp. 5895–5900.Google Scholar

[12] Azevedo, C., Poignet, P. and Espiau, B., "On line optimal control for biped robots," IFAC Proc. 35(1), 199–204 (2002). doi: 10.3182/20020721-6-ES-1901.00845.CrossRefGoogle Scholar

[13] Wieber, P.-B., " Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations," 2006 6th IEEE-RAS International Conference on Humanoid Robots (2006) pp. 137–142.Google Scholar

[14] Pratt, J. E. and Tedrake, R., "Velocity-based Stability Margins for Fast Bipedal Walking," In: Fast Motions in Biomechanics and Robotics (Springer, 2006) pp. 299–324.10.1007/978-3-540-36119-0$_1$4*CrossRef GoogleScholar*[15]*Joe, H.-M.andOh, J.-H., \Balancerecoverythroughmodelpredictivecontrolbasedoncapturepointdynamicsforbip* 10.1016*/j.robot*.2018.03.004.*CrossRef GoogleScholar*

[16] Pratt, J., Carff, J., Drakunov, S. and Goswami, A., " Capture Point: A Step Toward Humanoid Push Recovery," 2006 6th IEEE-RAS International Conference on Humanoid Robots (2006) pp. 200–207.Google Scholar

[17] Wight, D. L., Kubica, E. G. and Wang, D. W. L., “Introduction of the foot placement estimator: A dynamic measure of balance for bipedal robotics,” J. Comput. Nonlinear Dyn. 3(1) (2008).Google Scholar

[18] Erbatur, K., Okazaki, A., Obiya, K., Takahashi, T. and Kawamura, A., “ A Study on the Zero Moment Point Measurement for Biped Walking Robots ,” 7th International Workshop on Advanced Motion Control. Proceedings (Cat. No. 02TH8623) (2002) pp. 431–436.Google Scholar [19] Vundavilli, P. R., Sahu, S. K. and Pratihar, D. K., “Dynamically balanced ascending and descending gaits of a two-legged robot,” Int. J. Hum. Robot. 04(04), 717–751 (2007). doi: 10.1142/S0219843607001266.CrossRefGoogle Scholar [20] Vundavilli, P. R., Sahu, S. K. and Pratihar, D. K., “Online dynamically balanced ascending and descending gait generations of a biped robot using soft computing,” Int. J. Hum. Robot. 04(04), 777–814 (2007). doi: 10.1142/S0219843607001254.CrossRefGoogle Scholar [21] Vundavilli, P. R. and Pratihar, D. K., “Inverse dynamics learned gait planner for a two-legged robot moving on uneven terrains using neural networks,” Int. J. Adv. Intell. Paradig. 1(1), 80–109 (2008).Google Scholar

[22] Dekker, M. H. P., “Zero-moment point method for stable biped walking,” Eindhoven Univ. Technol. 2009, 1–15 (2009).Google Scholar

[23] Goswami, A., “Postural stability of biped robots and the foot-rotation indicator (FRI) point,” Int. J. Rob. Res. 18(6), 523–533 (1999). doi: 10.1177/02783649922066376.CrossRefGoogle Scholar

[24] Cannon, R. H. Dynamics of Physical Systems (Courier Corporation, 2003).Google Scholar [25] Schaefer, J. F. On the Bounded Control of Some Unstable Mechanical Systems (Stanford University, 1965).Google Scholar

[26] Xie, H., Zhao, X., Sun, Q., Yang, K. and Li, F., “A new virtual-real gravity compensated inverted pendulum model and ADAMS simulation for biped robot with heterogeneous legs,” J. Mech. Sci. Technol. 34(1), 401–412 (2020). doi: 10.1007/s12206-019-1239-4.CrossRefGoogle Scholar [27] Hemami, H., Weimer, F. and Koozekanani, S., “Some aspects of the inverted pendulum problem for modeling of locomotion systems,” IEEE Trans. Autom. Control 18(6), 658–661 (1973).CrossRefGoogle Scholar

[28] Gubina, F., Hemami, H. and McGhee, R. B., “On the dynamic stability of biped locomotion,” IEEE Trans. Biomed. Eng. 21(2), 102–108 (1974). doi: 10.1109/TBME.1974.324294.CrossRefGoogle ScholarPubMed

[29] Miyazaki, F. and Arimoto, S., “A control theoretic study on dynamical biped locomotion,” J. Dyn. Syst. Meas. Control 102(4), 233–239 (1980). doi: 10.1115/1.3149608.CrossRefGoogle Scholar

[30] Sangwan, V., Taneja, A. and Mukherjee, S., “Design of a robust self-excited biped walking mechanism,” Mech. Mach. Theory 39(12), 1385–1397 (2004). doi: 10.1016/j.mechmachtheory.2004.05.023.CrossRefGoogle Scholar

[31] van Zutven, P., Kosti´c, D. and Nijmeijer, H., “ On the Stability of Bipedal Walking ,” International Conference on Simulation, Modeling, and Programming for Autonomous Robots (2010) pp. 521–532.Google Scholar

[32] Chevallereau, C. and Aoustin, Y., “Optimal reference trajectories for walking and running of a biped robot,” Robotica 19(5), 557–569 (2001).CrossRefGoogle Scholar

[33] Westervelt, E. R. and Grizzle, J. W., “ Design of Asymptotically Stable Walking for a 5-Link Planar Biped Walker via Optimization,” Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), vol. 3 (2002) pp. 3117–3122.Google Scholar

[34] Chevallereau, C., Formal’sky, A. and Djoudi, D., “Tracking a joint path for the walk of an underactuated biped,” Robotica 22(1), 15–28 (2004).CrossRefGoogle Scholar

[35] Wang, K., Tobajas, P. T., Liu, J., Geng, T., Qian, Z. and Ren, L., “Towards a 3D passive dynamic walker to study ankle and toe functions during walking motion,” Rob. Auton. Syst. 115, 49–60 (2019). doi: 10.1016/j.robot.2019.02.010.CrossRefGoogle Scholar

[36] Goswami, A., Thuilot, B. and Espiau, B., Compass-Like Biped Robot Part I: Stability and Bifurcation of Passive Gaits (1996). INRIA, Jun. 1996. [Online]. Available at: https://hal.inria.fr/inria-00073701.Google Scholar

[37] Goswami, A., Espiau, B. and Keramane, A., “Limit cycles in a passive compass gait biped and passivity-mimicking control laws,” Auton. Robots 4(3), 273–286 (1997).CrossRefGoogle Scholar

[38] Spong, M. W. and Bullo, F., “Controlled symmetries and passive walking,” IFAC Proc. 35(1), 557–562 (2002).CrossRefGoogle Scholar

[39] Goswami, A., Thuilot, B. and Espiau, B., “A study of the passive gait of a compass-Like biped robot: Symmetry and chaos,” Int. J. Rob. Res. 17(12), 1282–1301 (1998). doi: 10.1177/027836499801701202.CrossRefGoogle Scholar

[40] Suzuki, S. and Furuta, K., “Enhancement of stabilization for passive walking by chaos control approach,” IFAC Proc. 35(1), 133–138 (2002). doi: 10.3182/20020721-6-ES-1901.00103.CrossRefGoogle Scholar

[41] Zheng, X.-D. and Wang, Q., “LCP method for a planar passive dynamic walker based on an event-driven scheme,” Acta Mech. Sin. 34(3), 578–588 (2018). doi: 10.1007/s10409-018-0749-0.CrossRefGoogle Scholar

[42] Vanderborght, B., Van Ham, R., Verrelst, B., Van Damme, M. and Lefeber, D., “Overview of the lucy project: Dynamic stabilization of a biped powered by pneumatic artificial muscles,” Adv. Robot. 22(10), 1027–1051 (2008).CrossRefGoogle Scholar

[43] Matsuoka, K., “Mechanisms of frequency and pattern control in the neural rhythm generators,” Biol Cybern. 56(5), 345–353 (1987).CrossRefGoogle ScholarPubMed

[44] Pandy, M. G., Anderson, F. C. and Hull, D. G., “A parameter optimiza- tion approach for the optimal control of large-scale musculoskeletal systems,” J. Biomech. Eng. 114(4), 450–460 (1992).CrossRefGoogle ScholarPubMed

[45] Zielińska, T., “Coupled oscillators utilised as gait rhythm generators of a two-legged walking machine,” Biol. Cybern. 74(3), 263–273 (1996).CrossRef-Google ScholarPubMed

[46] Or, J. and Takanishi, A., “ A Biologically Inspired CPG-ZMP Control System for the Real-Time Balance of a Single-Legged Belly Dancing Robot ,” 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), vol. 1 (2004) pp. 931–936.Google Scholar

[47] Al-Shuka, H. F. N., Allmendinger, F., Corves, B. and Zhu, W.-H., “Modeling, stability and walking pattern generators of biped robots: a review,” Robotica 32(6), 907–934 (2014).CrossRefGoogle Scholar

[48] Chevallereau, C., Bessonnet, G., Abba, G. and Aoustin, Y. Bipedal Robots: Modeling, Design and Walking Synthesis (John Wiley Sons, 2013).Google Scholar

[49] Shih, C.-L. and Gruver, W. A., “Control of a biped robot in the double-support phase,” IEEE Trans. Syst. Man. Cybern. 22(4), 729–735 (1992).Cross-RefGoogle Scholar

[50] Sano, A. and Furusho, J., “ Control of Torque Distribution for the BLR-G2 Biped Robot,” Fifth International Conference on Advanced Robotics ’Robots in Unstructured Environments, vol. 1 (1991) pp. 729–734. doi: 10.1109/ICAR.1991.240686.CrossRefGoogle Scholar [51] Choi, M. H. and Lee, B. H., “ A Real Time Optimal Load Distribution for Multiple Cooperating Robots,” Proceedings of 1995 IEEE International Conference on Robotics and Automation, vol.1 (1995) pp. 1211–1216.Google Scholar [52] Sonoda, N., Murakami, T. and Ohnishi, K., “ An Approach to Biped Robot Control Utilized Redundancy in Double Support Phase ,” Pro- ceedings of the IECON’97 23rd International Conference on Industrial Elec- tronics, Control, and Instrumentation (Cat. No. 97CH36066), vol. 3 (1997) pp. 1332–1336.Google Scholar

[53] Zhu, W.-H.. Virtual Decomposition Control: Toward Hyper Degrees of Freedom Robots, vol. 60 (Springer Science Business Media, 2010).10.1007/978-3-642-10724-5CrossRefGoogle Scholar

[54] Duindam, V. and Stramigioli, S., “Port-based control of a compass-gait bipedal robot,” IFAC Proc. 38(1), 471–476 (2005). doi: 10.3182/20050703-6-CZ-1902.00733.CrossRefGoogle Scholar

[55] Duindam, V. and Stramigioli, S., Modeling and Control for Efficient Bipedal Walking Robots: A Port-Based Approach, vol. 53 (Springer, 2008).Google Scholar

[56] Sangwan, V. and Agrawal, S. K., “Differentially flat design of bipeds ensuring limit cycles,” IEEE/ASME Trans. Mech. 14(6), 647–657 (2009).10.1109/TMECH.2009.2033593CrossRe Scholar

[57] Westervelt, E. R., Grizzle, J. W. and Koditschek, D. E., “Hybrid zero dynamics of planar biped walkers,” IEEE Trans. Autom. Control 48(1), 42–56 (2003).CrossRefGoogle Scholar

[58] Chevallereau, C., “Time-scaling control for an underactuated biped robot,” IEEE Trans. Robot. Autom. 19(2), 362–368 (2003).CrossRefGoogle Scholar

[59] Raibert, M., Tzafestas, S. and Tzafestas, C., “ Comparative Simulation Study of Three Control Techniques Applied to a Biped Robot,” Proceedings of IEEE Systems Man and Cybernetics Conference-SMC, vol.1 (1993) pp. 494–502.Google Scholar

[60] Park, I., Kim, J. and Oh, J., “Online Biped Walking Pattern Generation for Humanoid Robot KHR-3(KAIST Humanoid Robot - 3: HUBO),” 2006 6th IEEE-RAS International Conference on Humanoid Robots (2006) pp. 398–403. doi: 10.1109/ICHR.2006.321303.CrossRefGoogle Scholar

[61] Vukobratovic, M. and Juricic, D., “Contribution to the synthesis of biped gait,” IEEE Trans. Biomed. Eng. 16(1), 1–6 (1969). doi: 10.1109/TBME.1969.4502596.CrossRefGoogle Scholar

[62] Tzafestas, S., Raibert, M. and Tzafestas, C., “Robust sliding-mode control applied to a 5-link biped robot,” J. Intell. Robot. Syst. 15(1), 67–133 (1996). doi: 10.1007/BF00435728.CrossRefGoogle Scholar

[63] Peca, M., Sojka, M. and Hanzálek, Z., “SPEJBL – The biped walking robot,” IFAC Proc. 40(22), 63–70 (2007). doi: 10.3182/20071107-3-FR-3907.00010.CrossRefGoogle Scholar

[64] Mehmeti, X., “Adaptive PID controller design for joints of humanoid robot,” IFAC-PapersOnLine 52(25), 110–112 (2019). doi: 10.1016/j.ifacol.2019.12.456.CrossRefGoogle Scholar

[65] Kolathaya, S., “Local stability of PD controlled bipedal walking robots,” Automatica 114, 108841 (2020). doi: 10.1016/j.automatica.2020.108841.CrossRefGoogle Scholar

[66] Song, Z., Yi, J., Zhao, D. and Li, X., “A computed torque controller for uncertain robotic manipulator systems: Fuzzy approach,” Fuzzy Sets Syst. 154(2), 208–226 (2005). doi: 10.1016/j.fss.2005.03.007.CrossRefGoogle Scholar

[67] Markiewicz, B. R., Analysis of the Computed-Torque Drive Method and Comparision with the Conventional Position Servo for a Computer-Controlled Manipulator (1973). p. Technical Memorandum.Google Scholar

[68] Middletone, R. H. and Goodwin, G. C., “Adaptive Computed Torque Control for Rigid Link Manipulators,” 1986 25th IEEE Conference on Decision and Control (1986) pp. 68–73.Google Scholar

[69] Spong, M. W. and Vidyasagar, M. Robot Dynamics and Control (John Wiley Sons, 2008).Google Scholar

[70] Piltan, F., Mirzaei, M., Shahriari, F., Nazari, I. and Emamzadeh, S., “Design baseline computed torque controller,” Int. J. Eng. 6(3), 129–141 (2012).Google Scholar [71] Kurfess, T. R.. Robotics and Automation Handbook, vol. 414 (CRC press Boca, Raton, FL, 2005).Google Scholar

[72] Siciliano, B., Khatib, O. and Kröger, T. Springer Handbook of Robotics, vol. 200 (Springer, 2008).CrossRefGoogle Scholar

[73] Lewis, F. L., Jagannathan, S. and Ye¸sildirek, A., “Neural Network Control of Robot Arms and Nonlinear Systems,” In: Neural Systems for Control (Elsevier, 1997) pp. 161–211.CrossRefGoogle Scholar

[74] Albus, J. S., A new approach to manipulator control: The cerebellar model articulation controller (CMAC) (1975).CrossRefGoogle Scholar

[75] Lin, C.-M. and Chen, T.-Y., “Self-organizing CMAC control for a class of MIMO uncertain nonlinear systems,” IEEE Trans. Neural Networks 20(9), 1377–1384 (2009).CrossRefGoogle ScholarPubMed

[76] Guan, J., Hong, S., Kang, S., Zeng, Y., Sun, Y. and Lin, C.-M., “Robust adaptive recurrent cerebellar model neural network for non-linear system based on GPSO,” Front. Neurosci. 13, 390 (2019) https://doi.org/10.3389/fnins.2019.00390.CrossRefGoogle ScholarPubMed

[77] Zadeh, L. A., “Fuzzy sets,” Inf. Control 8(3), 338–353 (1965). doi: 10.1016/S0019-9958(65)90241-X.CrossRefGoogle Scholar

[78] Ahmadian, M., “Active Control of Vehicle Vibration,” In: V. S. B. T.-E. of Braun (Elsevier, Oxford, 2001) pp. 37–45, doi: 10.1006/rwvb.2001.0193.Google Scholar

[79] Hogan, N., “Impedance Control: An Approach to Manipulation,” 1984 American Control Conference (1984) pp. 304–313.Google Scholar [80] Hogan, N., Impedance control: An approach to manipulation: Part II—Implementation (1985).CrossRefGoogle Scholar

[81] Hogan, N. and Buerger, S. P., “Impedance and Interaction Control,” In: Robotics and Automation Handbook (CRC Press, 2018) pp. 375–398.Google Scholar

[82] Gatti, P. L. Applied Structural and Mechanical Vibrations: Theory, Methods and Measuring Instrumentation (CRC Press, 1999).CrossRefGoogle Scholar

[83] Sabanovic, A. and Ohnishi, K. Motion Control Systems (John Wiley Sons, 2011).CrossRefGoogle Scholar

[84] Lee, E. B. and Markus, L., Foundations of Optimal Control Theory (Minnesota Univ Minneapolis Center For Control Sciences, 1967).Google Scholar

[85] Garc´ıa, C. E., Prett, D. M. and Morari, M., “Model predictive control: Theory and practice—A survey,” Automatica 25(3), 335–348 (1989). doi: 10.1016/0005-1098(89)90002-2.CrossRefGoogle Scholar

[86] Magni, L. and Scattolini, R., “An overview of nonlinear model predic- tive control,” Lect. Notes Control Inf. Sci. 402(4), 107–117 (2010). doi: 10.1007/978-1-84996-071-7*7.GoogleScholar*

[87] Richalet, J., “Industrial applications of model based predictive control,” Automatica 29(5), 1251–1274 (1993).CrossRefGoogle Scholar

[88] Abu-Ayyad, M. and Dubay, R., “Real-time comparison of a number of predictive controllers,” ISA Trans. 46(3), 411–418 (2007).CrossRefGoogle ScholarPubMed

[89] Ren, Y. M., M. S. Alhajeri, J. Luo, S. Chen, F. Abdullah, Z. Wu and P. D. Christofides, “A tutorial review of neural network modeling approaches for model predictive control,” Comput. Chem. Eng. 165, 107956 (2022). doi: 10.1016/j.compchemeng.2022.107956.CrossRefGoogle Scholar

[90] Mandava, R. K. and Vundavilli, P. R., “Design and development of an adaptive-torque-based proportional-integral-derivative controller for a two-legged robot,” Soft Comput. 25(16), 10953–10968 (2021). doi: 10.1007/s00500-021-05811-4.CrossRefGoogle Scholar

[91] Golliday, C. L. Toward Development of Biped Locomotion Controls: Planar Motion Control of the Kneeless Biped Standing and Walking Gaits (The Ohio State University, 1975).Google Scholar [92] Hemami, H. and Golliday, C. L., “The inverted pendulum and biped stability,” Math. Biosci. 34(1-2), 95–110 (1977). doi: 10.1016/0025-5564(77)90038-4.CrossRefGoogle Scholar

[93] Miura, H. and Shimoyama, I., “Dynamic walk of a biped,” Int. J. Rob. Res. 3(2), 60–74 (1984). doi: 10.1177/027836498400300206.CrossRefGoogle Scholar [94] H u¨ rm u¨ zl u¨ , Y. and Moskowitz, G. D., “The role of impact in the stability of bipedal locomotion,” Dyn. Stab. Syst. 1(3), 217–234 (1986). doi: 10.1080/02681118608806015.Google Scholar

[95] Furusho, J. and Masubuchi, M., “A theoretically motivated reduced order model for the control of dynamic biped locomotion,” J. Dyn. Syst. Meas. Control 109(2), 155–163 (1987). doi: 10.1115/1.3143833.CrossRefGoogle Scholar
[96] Kajita, S. and Tani, K., “Study of Dynamic Biped Locomotion on Rugged Terrain-Theory and Basic Experiment,” Fifth International Conference on Advanced Robotics ’Robots in Unstructured Environments, vol. 1 (1991) pp. 741–746. doi: 10.1109/ICAR.1991.240688.CrossRefGoogle Scholar

[97] Kajita, S. and Tani, K., “ Study of Dynamic Biped Locomotion on Rugged Terrain-Derivation and Application of the Linear Inverted Pendulum Mode,” 1991 IEEE International Conference on Robotics and Automation Proceedings, vol. 2 (1991) pp. 1405–1411. doi: 10.1109/ROBOT.1991.131811,CrossRefGoogle Scholar [98] Furusho, J. and Sano, A., “Development of Biped Robot,” In: Advances in Psychology, vol. 78 (Elsevier, 1991) pp. 277–303. [Online]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0166411508607463 Google Scholar

[99] Kurcmatsu, Y., Katayama, O., Iwata, M. and Kitamura, S., “Autonomous Trajectory Generation of a Biped Locomotive Robot,” 1991 IEEE International Joint Conference on Neural Networks, vol. 3 (1991) pp. 1983–1988. doi: 10.1109/IJCNN.1991.170671.CrossRefGoogle Scholar [100] Vanderborght, B., Verrelst, B., Van Ham, R., Van Damme, M. and Lefeber, D., “Objective locomotion parameters based inverted pendulum trajectory generator,” Rob. Auton. Syst. 56(9), 738–750 (2008). doi: 10.1016/j.robot.2008.01.003.CrossRefGoogle Scholar [101] Latham, P., A Simulation Study of Bipedal Walking Robots: Modeling, Walking Algorithms, and Neural Network Control (1992). Doctoral Dissertations. Available at: https://scholars.unh.edu/dissertation/1698.Google Scholar [102] Kajita, S. and Tani, K., “Experimental Study of Biped Dynamic

Walking in the Linear Inverted Pendulum Mode," 1995 IEEE International Conference on Robotics and Automation, vol. 3 (1995) pp. 2885–2891. doi: 10.1109/ROBOT.1995.525693.CrossRefGoogle Scholar [103] Kajita, S. and Tani, K., "Experimental study of biped dynamic walking," IEEE Control Syst. 16(1), 13–19 (1996). doi: 10.1109/37.482132.Google Scholar

[104] Kajita, S. and Tani, K., "Adaptive gait control of a biped robot based on realtime sensing of the ground profile," Auton. Robots 4(3), 297–305 (1997).CrossRefGoogle Scholar [105] Fujimoto, Y. and Kawamura, A., "Simulation of an autonomous biped walking robot including environmental force interaction," IEEE Robot. Autom. Mag. 5(2), 33–42 (1998). doi: 10.1109/100.692339.CrossRefGoogle Scholar

[106] Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K. and Hirukawa, H., " The 3D Linear Inverted Pendulum Mode: A Simple Modeling for a Biped Walking Pattern Generation," RSJ/IEEE International Conference on Intelligent Robots and Systems, vol. 1 (2001) pp. 239–246. doi: 10.1109/IROS.2001.973365.CrossRefGoogle Scholar [107] Miyashita, T. and Ishiguro, H., "Human-like natural behavior generation based on involuntary motions for humanoid robots," Rob. Auton. Syst. 48(4), 203–212 (2004). doi: 10.1016/j.robot.2004.07.008.CrossRefGoogle Scholar [108] Wisse, M., Atkeson, C. G. and Kloimwieder, D. K., "Swing Leg Retraction Helps Biped Walking Stability," 5th IEEE-RAS International Conference on Humanoid Robots (2005) pp. 295–300. doi: 10.1109/ICHR.2005.1573583.CrossRefGoogle Scholar

[109] Kuo, A. D., "The six determinants of gait and the inverted pendulum analogy: A dynamic walking perspective," Hum. Mov. Sci. 26(4), 617–656 (2007). doi: 10.1016/j.humov.2007.04.003.CrossRefGoogle ScholarPubMed [110] Ghorbani, R., Wu, Q. and Wang, G. G., "Nearly optimal neural network stabilization of bipedal standing using genetic algorithm," Eng. Appl. Artif. Intell. 20(4), 473–480 (2007). doi: 10.1016/j.engappai.2006.09.007.CrossRefGoogle Scholar

[111] Ha, T. and Choi, C.-H., "An effective trajectory generation method for bipedal walking," Rob. Auton. Syst. 55(10), 795–810 (2007). doi: 10.1016/j.robot.2007.06.001.CrossRefGo Scholar

[112] Kajita, S., M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro and K. Yokoi, "Biped Walking Stabilization Based on Linear Inverted Pendulum Tracking," 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (2010) pp. 4489–4496. doi: 10.1109/IROS.2010.5651082.CrossRefGoogle Scholar

[113] A.M., B., A.S., K., Salehinia, Y. and Najafi, F., "An open loop walking on different slopes for NAO humanoid robot," Procedia Eng. 41, 296–304 (2012). doi: 10.1016/j.proeng.2012.07.176.Google Scholar [114] Al-Shuka, H. F. N., Corves, B. J., Vanderborght, B. and Zhu, W.-H., "Zero-moment point-based biped robot with different walking patterns," Int. J. Intell. Syst. Appl. 7(1), 31 (2014).Google Scholar [115] Kobayashi, T., Sekiyama, K., Hasegawa, Y., Aoyama, T. and Fukuda, T., "Unified bipedal gait for autonomous transition between walking and running in pursuit of energy minimization," Rob. Au-

ton. Syst. 103, 27–41 (2018). doi: 10.1016/j.robot.2018.02.005.CrossRefGoogle Scholar [116] Chevallereau, C., Razavi, H., Six, D., Aoustin, Y. and Grizzle, J., “Self-synchronization and self-stabilization of 3D bipedal walking gaits,” Rob. Auton. Syst. 100, 43–60 (2018). doi: 10.1016/j.robot.2017.10.018.CrossRefGoogle Scholar

[117] Bae, H. and Oh, J.-H., “Biped robot state estimation using compliant inverted pendulum model,” Rob. Auton. Syst. 108, 38–50 (2018). doi: 10.1016/j.robot.2018.06.004.CrossRefGoogle Scholar [118] De-Le´on-Gómez, V., Luo, Q., Kalouguine, A., Pámanes, J. A., Aoustin, Y. and Chevallereau, C., “An essential model for generating walking motions for humanoid robots,” Rob. Auton. Syst. 112, 229–243 (2019). doi: 10.1016/j.robot.2018.11.015.CrossRefGoogle Scholar [119] Jeong, H., Lee, I., Sim, O., Lee, K. and Oh, J.-H., “A robust walking controller optimizing step position and step time that exploit advantages of footed robot,” Rob. Auton. Syst. 113, 10–22 (2019). doi: 10.1016/j.robot.2018.12.003.CrossRefGoogle Scholar [120] Chang, L., Piao, S., Leng, X., He, Z. and Zhu, Z., “Inverted pendulum model for turn-planning for biped robot,” Phys. Commun. 42, 101168 (2020). doi: 10.1016/j.phycom.2020.101168.CrossRefGoogle Scholar [121] Kashyap, A. K. and Parhi, D. R., “Particle swarm optimization aided PID gait con- troller design for a humanoid robot,” ISA Trans. 114, 306–330 (2021). doi: 10.1016/j.isatra.2020.12.033.CrossRefGoogle ScholarPubMed [122] Ding, J., Xin, S., Lam, T. L. and Vijayakumar, S., Versatile Locomotion by Integrating An- kle, Hip, Stepping, and Height Variation Strategies (2021). Jun. 2021. [On- line]. Available at: https://www.research.ed.ac.uk/en/publications/versatile-locomotion-by-integrating-ankle-hip-stepping-and-height.Google Scholar

[123] Khan, A. T., Li, S. and Zhou, X., “Trajectory optimization of 5-link biped robot using beetle antennae search,” IEEE Trans. Circ. Syst. II Exp. Briefs, 1(10), 3276–3280 (2021). doi: 10.1109/TCSII.2021.3062639.Google Scholar

[124] Hemami, H. and Wyman, B., “Modeling and control of constrained dynamic systems with application to biped locomotion in the frontal plane,” IEEE Trans. Autom. Control 24(4), 526–535 (1979). doi: 10.1109/TAC.1979.1102105.CrossRefGoogle Scholar

[125] D.A.Bravo, M. and Rodas, C. F. R., “Design of a dynamic simulator for a biped robot,” Model Simul. Eng. 2021, 1–12 (2021). doi: 10.1155/2021/5539123.CrossRefGoogle Scholar [126] Caux, S. and Zapata, R., “Modeling and control of biped robot dynamics,” Robotica 17(4), 413–426 (1999). doi: 10.1017/S0263574799001411.CrossRefGoogle Scholar

[127] Vundavilli, P. R. and Pratihar, D. K., “Balanced gait generations of a two-legged robot on sloping surface,” Sadhana 36(4), 525–550 (2011). doi: 10.1007/s12046-011-0031-7.CrossRefGoogle Scholar

[128] Hernández-Santos, C., Rodriguez-Leal, E., Soto, R. and Gordillo, J. L., “Kinematics and dynamics of a new 16 DOF humanoid biped robot with ac- tive toe joint,” Int. J. Adv. Robot. Syst. 9(5), 190 (2012).CrossRefGoogle Scholar
[129] Gautam, R. and Patil, A. T., “Modeling and Control of Joint Angles of a Biped Robot Leg Using PID Controllers,” 2015 IEEE International

Conference on Engineering and Technology (ICETECH) (2015) pp. 1–5. doi: 10.1109/ICETECH.2015.7275042.CrossRefGoogle Scholar [130] Mandava, R. K. and Vundavilli, P. R., “Implementation of modified chaotic invasive weed optimization algorithm for optimizing the PID controller of the biped robot,” S ¯a dhan ¯a 43(5), 1–18 (2018).CrossRefGoogle Scholar [131] Mandava, R. K. and Vundavilli, P. R., “ Tuning of PID Controller Parameters of a Biped Robot Using IWO Algorithm ,” Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering (2018) pp. 90–94.Google Scholar

[132] Mandava, R. K. and Vundavilli, P. R., “Whole body motion generation of 18-DOF biped robot on flat surface during SSP DSP,” Int. J. Model. Identif. Control 29(3), 266–277 (2018).CrossRefGoogle Scholar

[133] Navaneeth, M. G., Sudheer, A. P. and Joy, M. L., “Contact wrench cone-based stable gait generation and contact slip estimation of a 12-DoF biped robot,” Arab J. Sci. Eng 47(12), 15947–15971 (2022). doi: 10.1007/s13369-022-06763-z.CrossRefGoogle Scholar

[134] Shih, C. L., Li, Y. Z., Churng, S., Lee, T. T. and Gruver, W. A. “Trajectory Synthesis and Physical Admissibility for a Biped Robot During the Single-Support Phase,” IEEE International Conference on Robotics and Automation (1990) pp. 1646–1652. doi: 10.1109/ROBOT.1990.126246.CrossRefGoogle Scholar [135] Shih, C.-L., Gruver, W. A. and Lee, T.-T., “Inverse kinematics and inverse dynamics for control of a biped walking machine,” J. Robot. Syst. 10(4), 531–555 (1993). doi: 10.1002/rob.4620100408.CrossRefGoogle Scholar

[136] Kljuno, E. and Williams, R. L., “Humanoid walking robot: Modeling, inverse dynamics, and gain scheduling control,” J. Robot. 2010, 1–19 (2010). doi: 10.1155/2010/278597 2010-06.CrossRefGoogle Scholar [137] Shih, C.-L., Zhu, Y. and Gruver, W. A., “Optimization of the Biped Robot Trajectory,” IEEE International Conference on Systems, Man, and Cybernetics (1991) pp. 899–903. doi: 10.1109/ICSMC.1991.169801.CrossRefGoogle Scholar [138] O’Flaherty, R., P. Vieira, M. Grey, P. Oh, A. Bobick, M. Egerstedt and M. Stilman, Kinematics and Inverse Kinematics for the Humanoid Robot HUBO2+ (Georgia Institute of Technology, 2013).Google Scholar

[139] Kumar, M. R., Lathan, L. S. and Vundavilli, P. R., “Dynamically balanced obstacle crossing gait generation of a biped robot using neural networks,” Int. J. Mech. Robot. Syst. 2(3-4), 232–253 (2015).CrossRefGoogle Scholar [140] Mandava, R. K. and Vundavilli, P. R., “Forward and Inverse Kinematic Based Full Body Gait Generation of Biped Robot,” 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) (2016) pp. 3301–3305.Google Scholar

[141] Mandava, R. K. and Vundavilli, P. R., “Study on Influence of Hip Trajectory on the Balance of a Biped Robot,” In: Emerging Trends in Electrical, Communications and Information Technologies (Springer, 2017) pp. 265–272.CrossRefGoogle Scholar [142] Kazemi, J. and Ozgoli, S., “Real-time walking pattern generation for a lower limb exoskeleton, implemented on the exoped robot,” Rob.

Auton. Syst. 116, 1–23 (2019). doi: 10.1016/j.robot.2019.02.012.CrossRefGoogle Scholar

[143] Oh, J., Sim, O., Jeong, H. and Oh, J.-H., “Humanoid whole-body remote-control framework with delayed reference generator for imitating human motion,” Mechatronics 62, 102253 (2019). doi: 10.1016/j.mechatronics.2019.102253.CrossRefGoogle Scholar [144] H.T., K., Balachandran, A. and Shah, S. V., “Optimal whole-body motion planning of humanoids in cluttered environments,” Rob. Auton. Syst. 118, 263–277 (2019). doi: 10.1016/j.robot.2019.04.004.Google Scholar

[145] Mandava, R. K. and Vundavilli, P. R., “An analytical approach for generating balanced gaits of a biped robot on stairs and sloping surfaces,” Int. J. Model. Identif. Control 33(1), 28–50 (2019).CrossRefGoogle Scholar

[146] Ceranowicz, A. Z., Planar Biped Dynamics and Control (1980). p. 1, Jun. 1980, [Online]. Available at: https://www.elibrary.ru/item.asp?id=7277031.Google Scholar

[147] Ceranowicz, A. Z. Planar Biped Dynamics and Control (The Ohio State University, 1979).Google Scholar

[148] Cambrini, L., Chevallereau, C., Moog, C. H. and Stojic, R., “Stable Trajectory Tracking for Biped Robots,” 39th IEEE Conference on Decision and Control, vol. 5 (2000) pp. 4815–4820. doi: 10.1109/CDC.2001.914690.CrossRefGoogle Scholar

[149] Chevallereau, C., Aoustin, Y. and Formal’sky, A., “Optimal Walking Trajectories for a Biped,” Proceedings of the First Workshop on Robot Mo- tion and Control. RoMoCo’99 (Cat. No.99EX353) (1999) pp. 171–176. doi: 10.1109/ROMOCO.1999.791071.CrossRefGoogle Scholar

[150] Townsend, M. A. and Seireg, A., “The synthesis of bipedal locomotion,” J. Biomech. 5(1), 71–83 (1972). doi: 10.1016/0021-9290(72)90020-6.CrossRefGoogle ScholarPubMed

[151] Formalsky, A. M., “Impulsive Control for Anthropomorphic Biped,” In: Theory and Practice of Robots and Manipulators (Morecki, A., Bianchi, G. and Jaworek, K., eds.) (Springer Vienna, Vienna, 1995) pp. 387–393. [Online]. Available at: http://link.springer.com/10.1007/978-3-7091-2698-1$_5$0.*CrossRef GoogleScholar*

[152] Yi, K. Y. and Zheng, Y. F., “Biped locomotion by reduced ankle power,” Auton. Robots 4(3), 307–314 (1997).CrossRefGoogle Scholar

[153] Sabourin, C. and Bruneau, O., “Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks,” Rob. Auton. Syst. 51(2-3), 81–99 (2005). doi: 10.1016/j.robot.2005.02.001.CrossRefGoogle Scholar

[154] Mandava, R. K. and Vundavilli, P. R., “Design of Near-Optimal Trajectories for the Biped Robot Using MCIWO Algorithm,” In: Soft Computing for Problem Solving (Springer, 2019) pp. 355–364.CrossRefGoogle Scholar

[155] Hemami, H. and Zheng, Y.-F., “Dynamics and control of motion on the ground and in the air with application to biped robots: dynamics and control of

motion," J. Robot. Syst. 1(1), 101–116 (1984). doi: 10.1002/rob.4620010107.CrossRefGoogle Scholar