



OPEN Standing balance of single-legged hopping robot model using reinforcement learning approach in the presence of external disturbances

S. Mohamad Hoseinifard & Majid Sadedel✉

In this scholarly investigation, the study focuses on scrutinizing the locomotion and control mechanisms governing a single-legged robot. The analysis encompasses the robot's movement dynamics pertaining to two primary objectives: executing jumps and sustaining equilibrium throughout successive jump sequences. Diverse concepts of this robot model have been scrutinized, leading to the introduction of a distinctive semi-active model devised for maintaining the robot's balance. The research involves an initial design for the robot model followed by the introduction of a multi-phase composite control system. As per the proposed model, the jumping action is facilitated through a four-link mechanism augmented by a spring, while balance preservation is achieved through the independent operation of two arms connected to the upper body. To address the successive jumps within the four-link mechanism, a multi-phase feedback controller is engineered. Additionally, a hybrid control strategy, incorporating the Deep Deterministic Policy Gradient algorithm (DDPG) along with a feedback controller, is proposed to sustain balance throughout the robot's contact and flight phases. The research outcomes, acquired through a series of comprehensive tests conducted within the Simulink simulator environment, demonstrate the robot's capacity to maintain balance over 80 consecutive jumps. The evaluations encompassed various simulated external disturbances, including 1- horizontal impacts on the upper body, 2- disparities in ground height, and 3- alterations in ground angle between consecutive steps. Notably, the findings showcase the robot's adeptness in maintaining balance despite an impact with an amplitude of 25 N for a duration of 0.1 seconds, as well as its resilience in managing ground height disparities up to 3 cm and ground angle variations of up to 3°.

Keywords Single-legged hopping robot, Standing balance, Reinforcement learning, Disturbance to legged robot

Hopping robots represent a category of robotic systems designed for locomotion through jumping. These robots typically incorporate mechanisms that produce potent propulsion to achieve considerable vertical elevation and navigate surfaces with the aid of guided mechanisms. The configuration of hopping robots can diverge based on their intended application. Certain hopping robots possess a singular leg or a set of legs, whereas others may exhibit multiple legs or even spheroidal bodies with internal mechanisms. These robots frequently employ internal sensors and control systems to observe their surroundings and adapt their jumping motions.

One of the primary obstacles in developing hopping robots is to attain equilibrium and regulation during the leaping motion. Researchers are endeavoring to optimize the robot's mechanisms, control algorithms, and feedback systems to ensure precise and steadfast jumps. Furthermore, energy efficiency is a crucial aspect, as the robot must effectively store and discharge energy to achieve recurrent jumping motions. Consequently, it is highly prevalent to employ passive elements for energy storage and regeneration. Hence, a model of hopping robots, named SLIP (Spring Loaded Inverted Pendulum), was introduced, enabling the robot to execute successive and discontinuous jumps through the utilization of the passive component in the leg that simulates animal motions that have inhomogeneous dynamical analysis.

Department of Mechanical Engineering, Tarbiat Modares University, Tehran, Iran. ✉email: majid.sadedel@modares.ac.ir

Therefore, investigation into the development of jumping robots commenced during the 1980s at MIT University, where a three-dimensional jumping robot utilizing a hydraulic actuator was constructed¹. In², drawing upon the framework presented in¹, a non-linear controller was proposed to track a sequence of waypoints within a confined trajectory. In³, three generations of jumping automata were presented, which executed the jumping motion in a sequential and discontinuous manner. In⁴, a mathematical framework was introduced, enabling the derivation of closed-form solutions to determine a “SLIP-Spring Loaded Inverted Pendulum” model. In⁵, a technique referred to as “Force-Bit Manipulation” was presented, tailored for the fine-grained manipulation of joint torque in the robot’s low-level control. In⁶, scholars worked on locomotion of “Stumpy” robot which is accomplished through the application of sinusoidal and alternating forces to the ground, engendering frictional interactions with the ground plane. Subsequently, through the incorporation of artificial intelligence techniques, within the context of the study⁷, a cascade control approach was employed to formulate joint positioning for a pneumatic-actuated hopping robot. Notably, the manipulation of control coefficients was fine-tuned utilizing a RBF neural network. In⁸, an introduced control strategy termed “Hybrid Feedback Control” was delineated, featuring a dual-core control architecture aimed at regulating the foot’s landing angle and regenerating energy dissipation. This technique was instantiated on the SLIP model, adhering to predefined waypoints to ensure consistent speed in a 2D plane. Moreover, in⁹, an analogous approach was applied to a 3D model, thereby accounting for the robotic foot’s mass and devising an appropriate landing angle to ensure accurate adherence. In¹⁰, a pneumatic actuator employing dynamic elastic characteristics is utilized, controlled by the “PD” controller to furnish the requisite propulsive force for the system. In¹¹, a parallel mechanism for robot movement was introduced, whose control system included a Jacobian matrix to observe the state and a cascade control system called phase control was designed. In¹², a basic planar model was devised for the bipedal robot named “Kong”, capable of sustaining equilibrium through ankle joint adjustments within the limited upper body mobility.

In¹³, researchers introduced an approach using predictive control to regulate the velocity of a humanoid robot in both the horizontal and vertical dimensions, the robot guidance and control system consists of three parts. In¹⁴, a control method of the robot was checked to maintain the center of mass in a permissible range, and by considering the SLIP between the robot’s sole and the ground, the problem became closer to reality. In¹⁵, a control mechanism was introduced to regulate the leg position of a single-legged robot with a closed mechanism, which guided the robot in three different movement phases based on impedance control. In¹⁶, the researchers introduced adaptive control for the movement of the jumping robot’s joints, which was compressed by the rotating mass of the spring in the robot’s leg. In¹⁷, a robot based on the SLIP model was introduced, which can move its center of mass away from the leg by changing the angle of each arm. It can also design and follow the landing angle of the foot during flight with Backstepping control. Also, in¹⁸, a theory was presented for the stable landing of the biped robot, which was possible by accurately solving the dynamics of the robot model. In¹⁹, a mechanism similar to¹⁶ was used, where the position of the robot’s joints was designed by adaptive control. Simultaneously, lower level of control, the feedback control mechanisms were designed to impart dynamic load upon the robotic joints in accordance with the pre-specified trajectory. In²⁰, the landing angle of the foot is regulated through the resolution of inverse kinematics and energy equations, employing a proportional controller during the flight phase. Following this, in²¹, a fuzzy agent was proposed for the optimal adjustment of the control parameters introduced in²⁰. In^{22,23}, a method featuring real-time adaptability is presented for bipedal robot locomotion on uneven terrain, under the premise that the surface height is subject to change. In²⁴, an investigation was conducted into the management of a two-dimensional robot model employing SLIP. Within this study, a fuzzy agent was meticulously crafted to fine-tune the controller coefficients. In²⁵, an exploration was undertaken to configure the trajectory for the joints of a hopping robot named Salto-IP. In²⁶, an investigation was conducted on TTI-Hopper, a uni-pedal robotic platform with a biomechanically inspired human leg-like structure.

With the introduction of reinforcement learning algorithms, the development of mobile robot maneuvers began with the help of AI agents, which were widely used in the discussion of path design and online control. In the same direction in²⁷, Tutsoy et al. designed a controller to maintain the static balance of the robot by synthesizing the solution of inverse kinematic equations and implementing the reinforcement learning algorithm on a 3D humanoid robot with 12 degrees of freedom. In²⁸, the analysis of control and standing balance delineates into two distinct tiers, high-level control and low-level control. In the high-level control phase, the learning agent utilizes the reinforcement learning methodology to dispatch instructions to the low-level control layer. Furthermore, on occasion, basic reinforcement learning algorithms have been employed for the advancement of legged robot locomotion²⁹. However, their practical utility has been hindered by inherent constraints, particularly their limitations in making real-time decisions within a continuous-time and continuous workspace environment. Consequently, deep reinforcement learning, with its capacity to train neural networks seamlessly in continuous-time scenarios, has garnered significant research focus.

In³⁰, a pioneering approach was introduced to implement the reinforcement learning algorithm on the real mechanism of robots. This approach explicitly accounted for uncertainties in the robot’s mechanical structure and the presence of noise in the observational data during simulation.

Previous research primarily focused on simplistic modeling by applying controllers to models that lacked significant robustness to maneuvers, functioning solely on a simplified framework. Consequently, unpredictable factors could adversely impact the robot’s performance. Additionally, numerous articles concentrated exclusively on two-dimensional models or restricted the upper body movements to the vertical axis.

Regarding control, it can also be mentioned that in recent research, controlling robots with two conventional and intelligent approaches has been very popular. In this section, both groups of approaches were examined with examples. Regarding classical approaches, solving problems is generally based on accurate modeling of system dynamics, and therefore the accuracy of the response and efficiency of the system depends on the accuracy of the model definition and the analytical solution with which the design work is to be done.

So, given the intricate nature of analyzing the dynamics and kinematics of hopping robots and the segmentation of their motion maneuvers into distinct phases during both the stance and flight, it is prudent to employ intelligent control mechanisms in formulating trajectory designs and controlling the degrees of freedom, to achieve enhanced precision and robustness in executing maneuvers. Consequently, this approach allows us to circumvent the intricacies associated with analysis, which frequently entail numerous simplifications and assumptions. And at different levels of control, an AI agent can be trained to perform missions.

This research presents an innovative model designed to sustain the balance of a robot capable of executing repeated jumps, based on an analysis of prior research^{6,27–30}, and describes the development of a SLIP-based robot model capable of autonomous 3D jumping, leveraging the foundational robot model introduced in³¹. By combining the concept of reinforcement learning and feedback control, a control system is introduced that allows the robot to apply the required momentum to jump and select the appropriate step angle (through arm movements) to prevent the robot from falling.

The proposed model, as described in Sect. 2, incorporates a passive leg mechanism to accumulate kinetic energy with each step, enabling continuous and energy-efficient. Furthermore, in order to execute the maneuver of standing balance, a collaborative control system comprising the DDPG algorithm and PD control is employed, as elucidated in Sect. 3. Additionally, within Sect. 4, the outcomes of the simulations are showcased, followed by a comprehensive examination and summarization of the findings in Sect. 5.

Model description

The robotic leg model comprises a spring-based system integrated with an active four-link mechanism³¹. Upon each impact of the robot with the ground, the spring undergoes compression and subsequently discharges its stored energy, propelling the robot upward. However, to counterbalance the energy dissipation occurring with each jump and maintain a consistent jumping height, the system depends on an active actuator. Upon landing, this actuator, in the form of a motor, retracts the mechanism and compresses the spring. Also, to establish balance collapsing within the robotic system, counterweights are strategically positioned on either side of the arms to enable dynamic adjustment of the leg angles (Fig. 1). In summary, this system comprises three actuators that necessitate precise control: one motor for sustaining the robot's jumping motion, and the remaining two motors for maintaining the robot's stability during its jumps.

The robot model has been simulated using MATLAB software, leveraging the Simscape tools within the Simulink library. By the modeling process, the following presumptions have been taken into account:

- All aerodynamic forces are ignored with the gravitational acceleration of $9.81 \frac{m}{s^2}$.
- For all joints and connections, linear damping and stiffness are included.
- The normal contact force between the foot and the ground is considered as a 3rd order non-linear collision model.
- The tangential interaction at the interface between the foot and the ground is regarded as a form of linear frictional contact.

The model introduced in this investigation bears a resemblance to the previously developed SLIP model. The kinematic equations governing this analogous model have been extracted independently for both flight and stance phases, as detailed in³¹. The difference between the models lies in the replacement of the conventional 4-point-to-ground contact with a spherical attachment at the robot's toe. This alteration imparts increased agility to the robot but simultaneously poses challenges in terms of stability maintenance.

Owing to the nature of hopper robot locomotion and the distinct dynamics characterizing the robot during its flight and stance phases, the formulae elucidating the robot's jumping behavior are presented in a bifurcated manner. Furthermore, the model about the interaction between the robot's toe and the ground is delineated in Appendix A.

The implemented mechanism featuring four degrees of freedom, one of which operates passively, enables the robotic system to maneuver within the plane by manipulating its arms within the respective planes (Fig. 2). Consequently, the robot can dynamically adjust the orientation of its leg and body by exerting torque at the

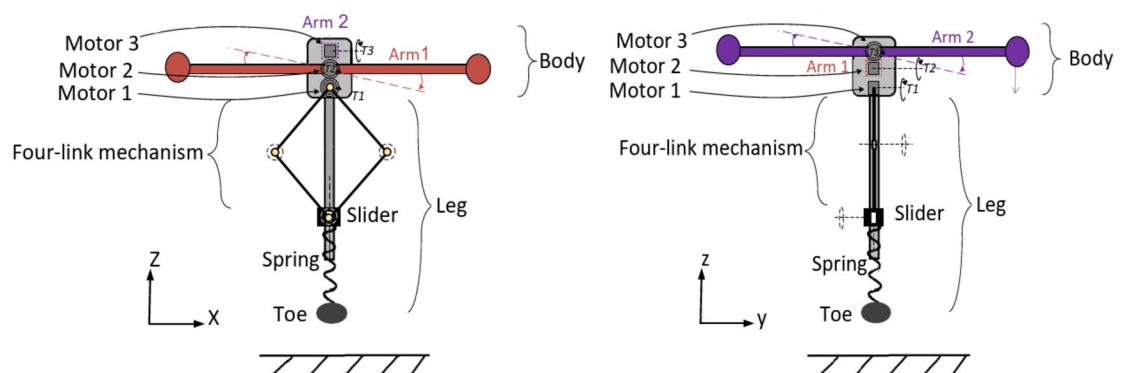


Fig. 1. Side views of robot model.

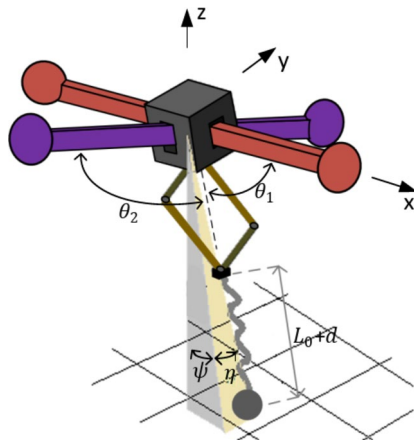


Fig. 2. Robot model in 3D space.

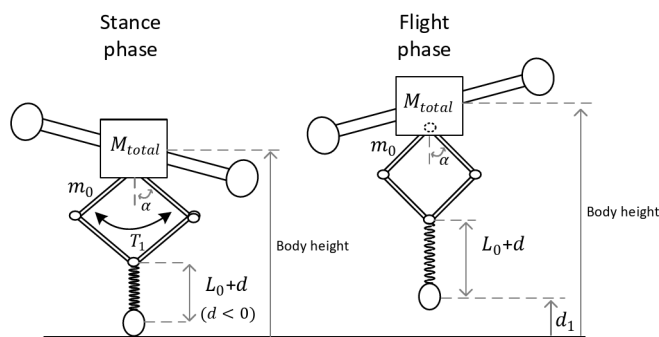


Fig. 3. Robot's stance and flight position.

upper body joints, facilitating horizontal movement on the plane while ensuring optimal leg landing angles or counterbalancing external forces and disturbances.

The crucial aspect of modeling a hopping robot mechanism lies in ensuring a substantial mass distribution in the robot's upper body (comprising the arms) rather than the leg. It is imperative that, during the flight phase, the torque exerted by the arm actuators results in minimal angular changes in the arms³¹.

This constraint increases the accuracy in the leg's angle orientation, and operational range and mitigates the destabilizing effect caused by excessive arm angular deviation on the robot's balance. For this reason, the mass of each robot arm is assumed to be approximately twenty times that of the robot's toe.

To simplify the robot's motion for executing two jumping tasks while ensuring balance, the motion system has been analyzed to enable independent control of both actions. In the proposed model, a four-link mechanism is employed for jumping, while the arms are solely manipulated to adjust the angle of the robot's legs.

In this manner, a four-link mechanism has been incorporated into the middle segment of the robot's leg to offset the energy expenditure during each step, by imparting supplementary momentum from the upper body to the spring upon the robot's landing during the contact phase (Fig. 3).

Control methodology

In this section, the proposed control method of balancing will be discussed. The approach for implementing motion and ensuring the robot's balance relies on classical feedback control and the deep deterministic policy learning algorithm "DDPG". The rationale behind employing this artificial intelligence-based methodology is to enhance the robot's robustness during jumping maneuvers and balance maintenance.

The deep deterministic policy gradient algorithm represents a reinforcement learning technique amalgamating concepts from policy-based and value-based strategies. Specifically engineered to address challenges in continuous action spaces, "DDPG" adopts an actor-critic framework comprising two neural networks: an actor network and a critic network. The actor network discerns the policy mapping states to actions, whereas the critic network gauges the value function, thereby assessing the efficacy of chosen actions.

Owing to their aptitude for continuous-time learning, actor-critic algorithms play a pivotal role in enhancing received or command signals, mapping, strategy selection, and the generation of control commands within the realm of robotics. To leverage these capabilities, a control framework is devised by amalgamating classical control principles with trained networks through a reinforcement agent. Within these mechanisms, a phase

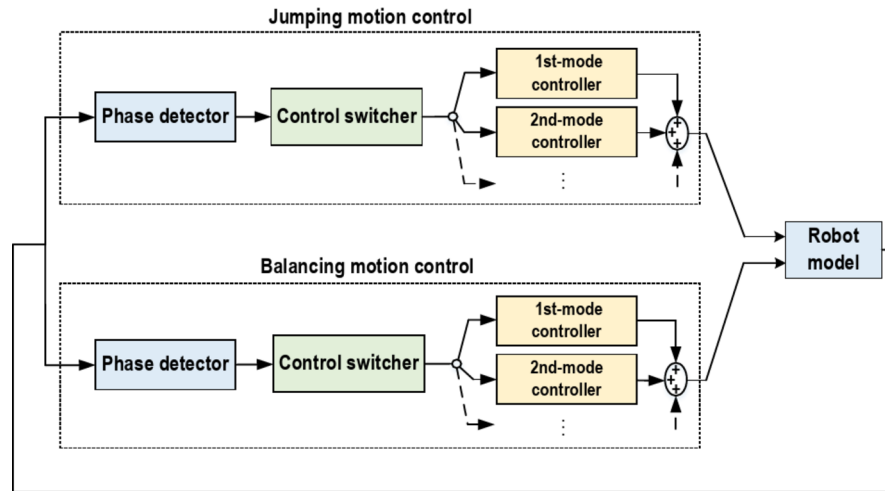


Fig. 4. Control system diagram.

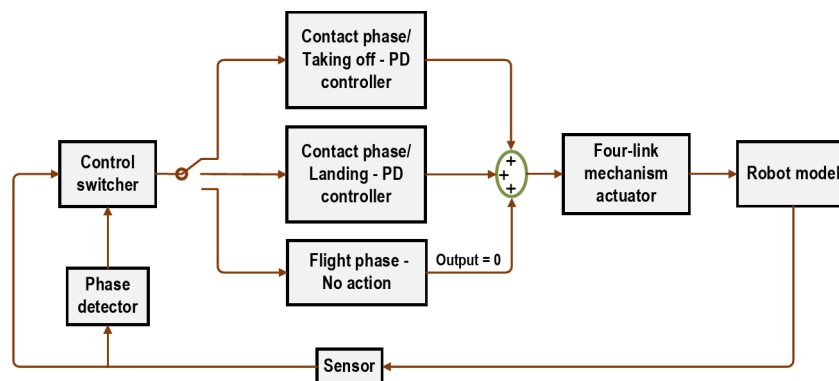


Fig. 5. Jumping control diagram.

detector is employed to assess the robot's current situation by integrating feedback from its various components, including position, speed, and forces (Fig. 4).

Jumping control

In accordance with the proposed framework (Fig. 5), the control input is administered to the drive actuator of the four-link mechanism to facilitate its opening and closing within a specific range. This work is estimated by the feedback controller “PD”. It is imperative to observe that the error signal is derived from the mechanism's angle and the spring's length (Eq. 1).

Therefore, as the spring undergoes a length change (compression) in the contact/landing phase and the four-link mechanism is about to change its shape, the operator does not allow the mechanism to retract too much. And when the compression reached the maximum possible and the movement phase changed from the contact/landing phase to the contact/request phase, now by changing the control coefficients, the robot will collect the mechanism more intensively. In this way, the energy consumption caused by the impact of the foot and the ground and the damping of the joints are compensated.

$$e_1 = k_1 \alpha - k_2 (L_0 + d) \quad (1)$$

Equation (1) delineates the error signal of the controller, employing the variation in the length of the linear spring and the contraction rate of the mechanism as feedback inputs. This signal elucidates the disparity between the folding behavior of the four-link mechanism and that of the spring's compression.

By selecting:

$$k_1 = \frac{4k_2 (L_0)}{\pi} \quad (2)$$

As long as the four-link angles $\alpha_0 = \pi/4$ radians and the spring remains undeformed ($d = 0$), the signal error attains zero. Consequently, with the spring's free length, the ratio of two coefficients, k_1 and k_2 , remains

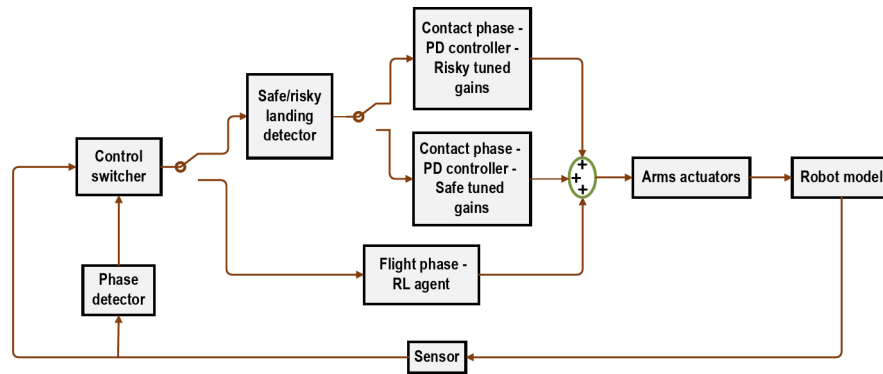


Fig. 6. Balancing control diagram.

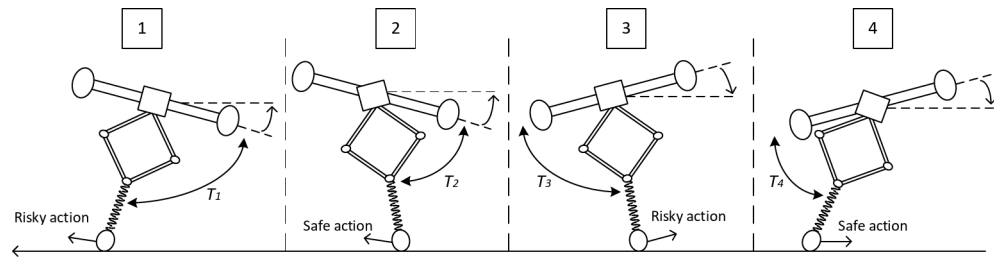


Fig. 7. The robot's four possible landing state.

constant (Eq. 2). The actuator 1 enables the robot to perform jumping movements by controlling the four-link mechanism. T_1 is exerted by the PD controller, which is adjusted based on two distinct motion states.

$$T_1 = \begin{cases} k_{p1-1}e_1 + k_{d1-1}\dot{e}_1 & d_1 \ll 0 \text{ and } \dot{d} \ll 0 \\ k_{p1-2}e_1 + k_{d1-2}\dot{e}_1 & d_1 \ll 0 \text{ and } 0 < \dot{d} \\ 0 & 0 < d_1 \end{cases} \begin{matrix} \text{Contact - Landing} \\ \text{Contact - Taking off} \\ \text{Flight} \end{matrix} \quad (3)$$

In the contact-landing phase, Eq. (3) demonstrates that the PD controller is utilized with varying gains, unlike in the contact-taking off phase. Additionally, the four-link mechanism remains inactive when the robot's toe is not in contact with the ground. The PD factors, which are adjusted through simulation tests³¹, influence the robot's vertical jump height. By taking into account the suggested robot's mechanism, this approach minimizes conflicts in managing and regulating both jumping and horizontal movements.

Balancing control

Ensuring the balance necessitate (1) precise alignment of the landing leg angle and (2) precise adjustment of the upper body angle upon leg-ground contact. Previous scholarly investigations have examined the manipulation and regulation of foot angle during landing in two-dimensional robotic models, focusing on scenarios where the upper body mass is concentrated at a singular point, as indicated by the findings in [1 and 5]. However, when accounting for the arms of the upper body in two distinct planes, x-z and y-z, it becomes imperative to employ accurate torque application during the robot's ascent from the ground.

In accordance with Fig. 6, to preserve balance during the leg-ground contact phase, the 'PD' controller readies the robot for the flight phase by resetting the arms to a horizontal position, parallel to the ground. Additionally, a landing position detector is positioned in front of the controller. If the contact angle between the foot and the ground is such that applying torque to move the arm might lead to the foot detaching and the robot falling, the operator modulates the output with a reduced gain during the contact phase. As illustrated in Fig. 7, in modes 1 and 3, excessive torque applied to adjust the arms poses a risk of the foot disengaging from the ground.

Therefore, the estimation of the torque of actuators 2 and 3 is according to the following equations:

$$T_2 = \begin{cases} k_{p2-1}\theta_1 + k_{d2-1}\dot{\theta}_1 & \frac{\pi}{4} - (\theta_1 + \psi)\psi \ll 0 \\ k_{p2-2}\theta_1 + k_{d2-2}\dot{\theta}_1 & 0 \ll \frac{\pi}{4} - (\theta_1 + \psi)\psi \end{cases} \begin{matrix} \text{Safe action} \\ \text{Risky action} \end{matrix} \quad (4)$$

$$T_3 = \begin{cases} k_{p3-1}\theta_2 + k_{d3-1}\dot{\theta}_2 & \frac{\pi}{4} - (\theta_2 + \eta)\eta \ll 0 \\ k_{p3-2}\theta_2 + k_{d3-2}\dot{\theta}_2 & 0 \ll \frac{\pi}{4} - (\theta_2 + \eta)\eta \end{cases} \begin{matrix} \text{Safe action} \\ \text{Risky action} \end{matrix} \quad (5)$$

1	V_{b-x}	Velocity of the center of the body in the x-coordinate direction
2	V_{b-y}	Velocity of the center of the body in the y-coordinate direction
3	V_{b-z}	Velocity of the center of the body in the z-coordinate direction
4	V_{t-x}	Velocity of the toe in the x-coordinate direction
5	V_{t-y}	Velocity of the toe in the y-coordinate direction
6	V_{t-z}	Velocity of the toe in the z-coordinate direction
7	P_{t-z}	Position of the center of the toe in the z direction
8	W_{b-z}	Rotational velocity of the center of the body in the z direction
9	A_{l-x}	Rotational position of the robot's leg around the X-axis of fixed coordinates
10	A_{l-y}	Rotational position of the robot's leg around the Y-axis of fixed coordinates
11	W_{l-x}	Rotational velocity of the robot's leg around the x axis
12	W_{l-y}	Rotational velocity of the robot's leg around the y axis
13	A_{4l}	Position of the four-link mechanism
14	W_{4l}	Folding velocity of the four-link mechanism
15	A_{a1-y}	Rotational position of arm 1 around the Y axis of fixed coordinates
16	A_{a2-x}	Rotational position of arm 2 around the X axis of fixed coordinates
17	F_n	Vertical ground reaction force at the toe
18	P_s	Spring deformation
19	V_s	Deformation rate of the spring
20	T_1	Torque of actuator 1
21	T_2	Torque of actuator 2
22	T_3	Torque of actuator 3
23	A_{a1}	Joint position of arm 1
24	A_{a2}	Joint position of arm 2

Table 1. Observation space.

1	T_2	Torque of actuator 2
2	T_3	Torque of actuator 3

Table 2. Action space.

In the subsequent phase, as the foot detaches from the ground during the flight phase, driving the motion of the arms and the lower body in three-dimensional space, is executed using the “DDPG” algorithm. During this phase, the reinforcing agent endeavors to uphold balance while striving to maximize the cumulative reward as defined by the reward function (Eq. 6).

The observation space of the reinforcement agent encompasses 24 signals derived from both dynamic and kinematic indicators of the robot, listed in Table 1. These signals, which serve as inputs for the reinforcement controller, offer a comprehensive real-time understanding of the robot's situation to the agent.

In addition, the balancing movement is dictated by the actuators that manipulate arm 1 and arm 2 (actuators 2 and 3), the action space of the reinforcement learning agent is stated in Table 2.

A critical determinant influencing the agent's rapid and accurate learning process is the specification of the reward function. This function must encompass all definitions of the robot's appropriate or inappropriate performance, necessitating extensive feedback characterized by precise communication. It is worth noting, however, that the intricate nature of this function can create ambiguity for the agent when selecting the optimal policy. In this study, given the time-sensitive nature of executing the balance maneuver, this parameter is incorporated into the reward function with a nonlinear impact. Specifically, as the duration of each episode lengthens, the robot accrues rewards at an escalating rate.

The reward function is defined as follows:

$$R = 60t((0.6 - |A_{l-x}|) + (0.6 - |A_{l-y}|) - |A_{a1}| - |A_{a2}| + 0.3F_n + (0.35 - 5|T_2|) + (0.35 - 5|T_3|) + 8|A_{a1-y}| + 8|A_{a2-x}|) \quad (6)$$

As per the reward function specifications, the agent is incentivized over the extended duration to:

- (1) Extend the duration of walking on a vertical leg within each episode.
- (2) Minimize excessive changes in arm angles.
- (3) Maintain a horizontal orientation of the hands.
- (4) Make more contact between the foot and the ground so that the greater the force (i.e. the angle between the foot and the ground is 90°), the more reward will be obtained.

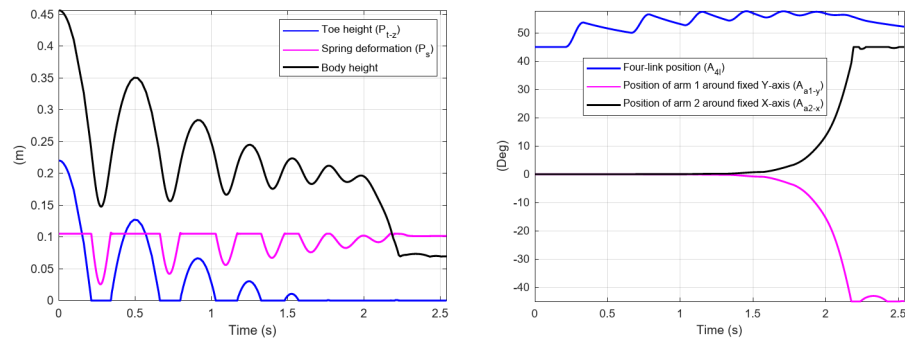


Fig. 8. The robot height, spring displacement, and the robot joint position with no action.

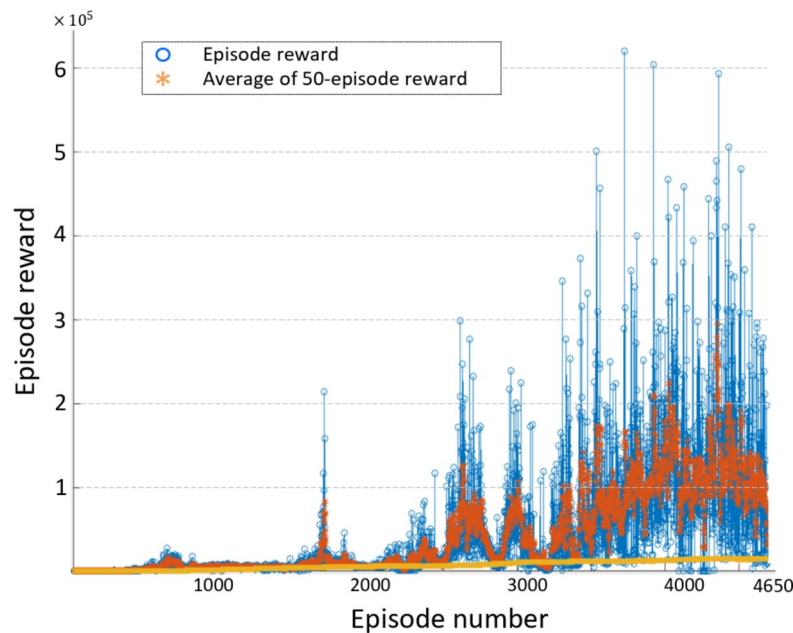


Fig. 9. Rewards earned in the training process.

- (5) Avoid excessive torque application during episodes, which is also aligned with the objectives outlined in item 2.

Simulation and results

Model verification

To corroborate the precision of the robot model implemented in simulation software, an assessment is conducted in which the robot is dropped from a height of 22 cm, and the sensors on the robot relay its movement data. This approach is adopted due to the robot being simulated in a pseudo-physical environment.

Hence, initially, the mechanical parameters of the robot are presented based on the data provided in Appendix B.

Furthermore, there exists a constraint on the rotational movement of the robot's arms, restricting them from exceeding a 90° rotation. Upon reaching this limit, the arms experience a substantial increase in stiffness and damping, compelling the arm to revert back within the predefined range ($k_{limit} = 100000 \text{ N.m/deg}$ & $c_{limit} = 100000 \text{ N.m/deg}$).

Now, assuming $T_1 = T_2 = T_3 = 0$, the robot must fall to the ground after several jumps. According to the Fig. 8, the robot falls to the ground in 2.6 s.

Motion with no disturbances

According to the model and control methodology description, the simulation in question was executed by computation of control parameters. These values were fine-tuned employing the traditional Ziegler-Nichols methodology (for more details see Appendix C). Moreover, the values pertaining to reinforcement learning and the RNN network parameters are delineated in Appendix D. By configuring the reinforcement learning parameters, the reward diagram depicting the learning process is derived as illustrated in Fig. 9.

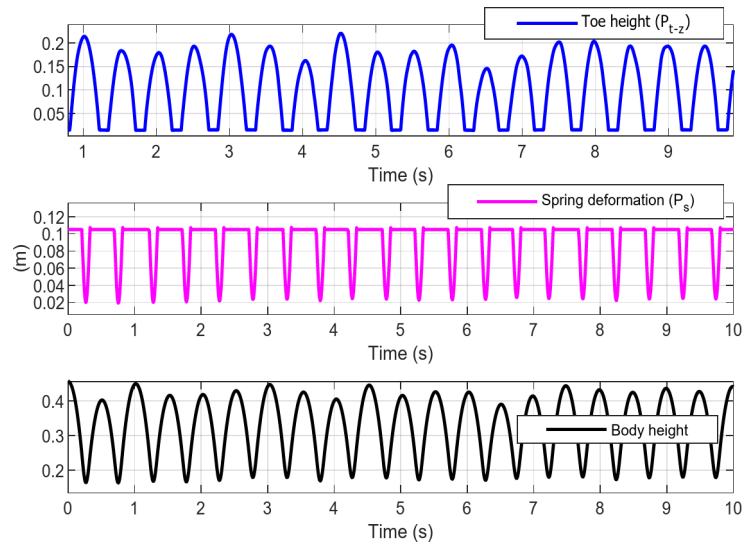


Fig. 10. The robot height and spring displacement.

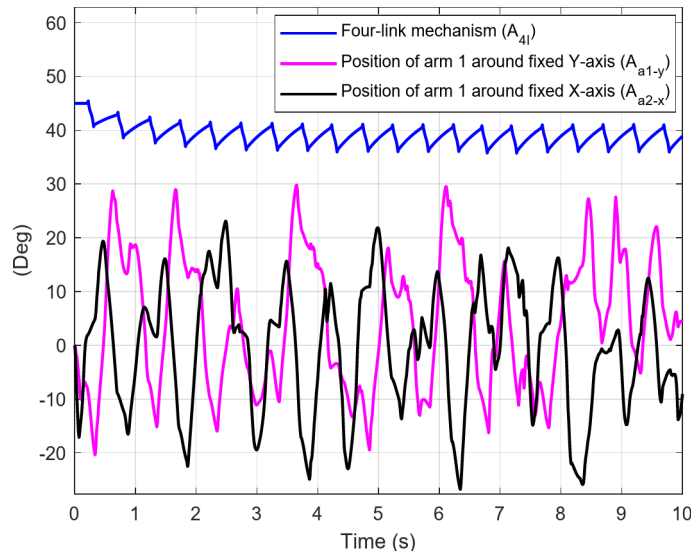


Fig. 11. The Joints position of the robot.

Across 4650 episodes, the robot consistently maintained an average cumulative reward of approximately 100,000 over 50 episodes. Moreover, evident from the training process was a notably elevated reward distribution in the concluding episodes, indicating the robot's adeptness in sustaining balance and accumulating greater experiential knowledge to augment cumulative rewards. The subsequent diagram illustrates the temporal variations in the robot's foot-to-ground center height, linear spring contraction within the leg, and the overall body height throughout a 10 s simulation.

Figure 10 illustrates the succession of leaps executed by the robot, with the jump's altitude consistently hovering around 42 cm and the dampening resulting from the foot-to-ground impact and other interconnections' dampening being compensated by the controller. Additionally, Fig. 11 depicts the ability of joint 1 to propel the robot into the air after each landing, achieved by modulating the angle within a 4° range.

It is noteworthy that the torques T_2 and T_3 were generated by the reinforcement learning factor at a frequency of 100 Hz within the range of $[-1,1]$ N.m (Fig. 12).

Additionally, Figs. 13, 14 and 15 illustrates the simulation outcome of the robot's behavior over 40 s, wherein the robot traversed a path randomly spanning approximately 8 m in order to keep its balance. In Fig. 13, the trajectory points of the robot joints, confined within a specified region, illustrate the stability of the proposed control method.

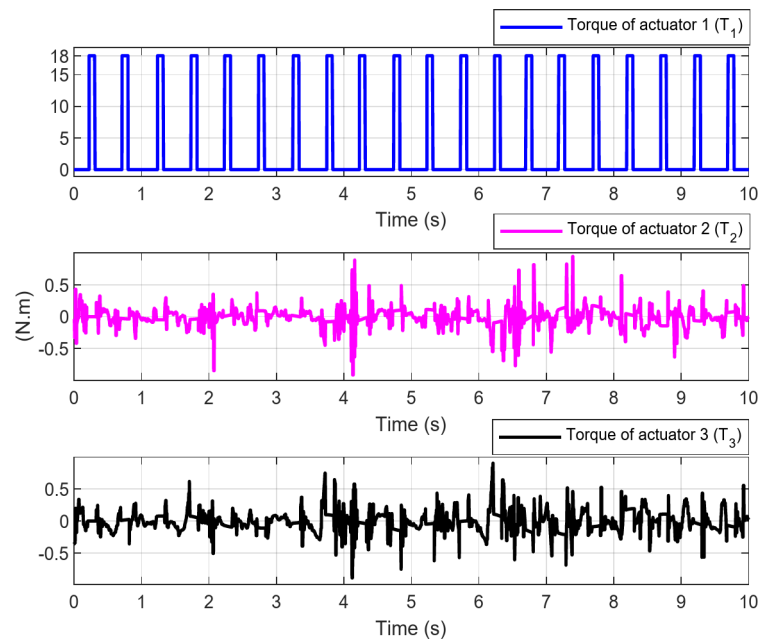


Fig. 12. Torques of actuators applied.

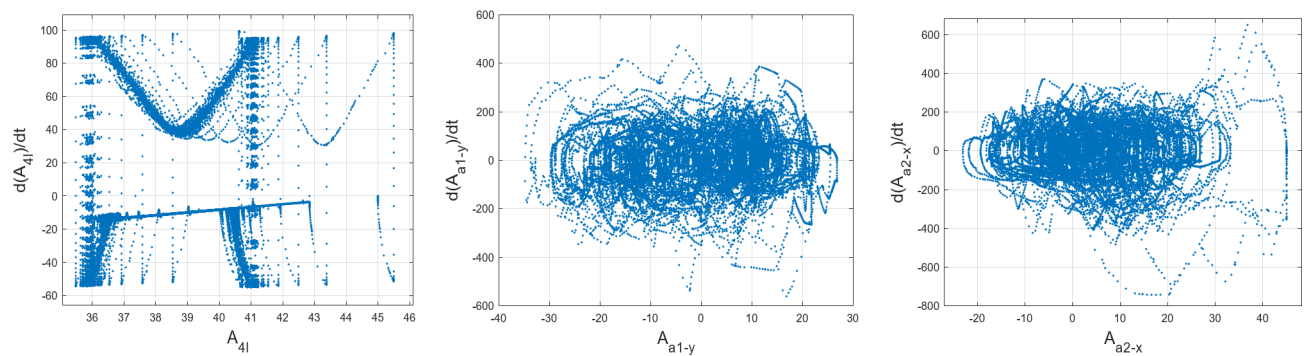


Fig. 13. Poincaré section pertaining to the joints of the robot.

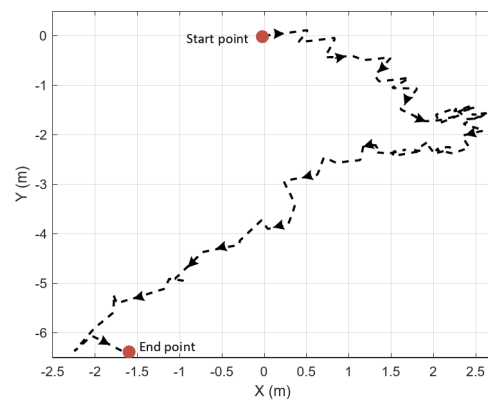


Fig. 14. The path taken by the robot.

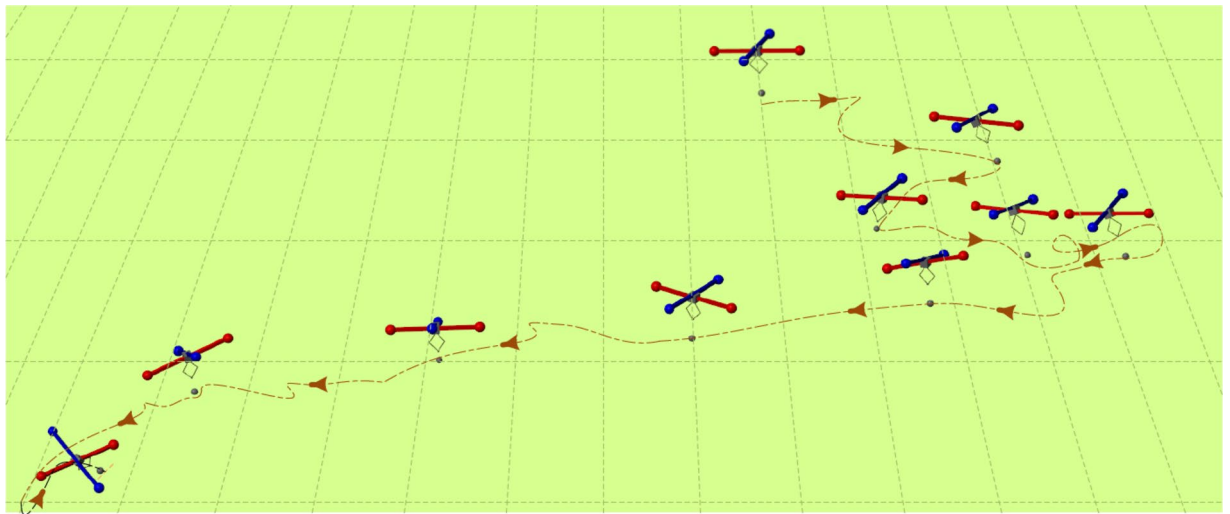


Fig. 15. The robot's shots in the simulation environment.

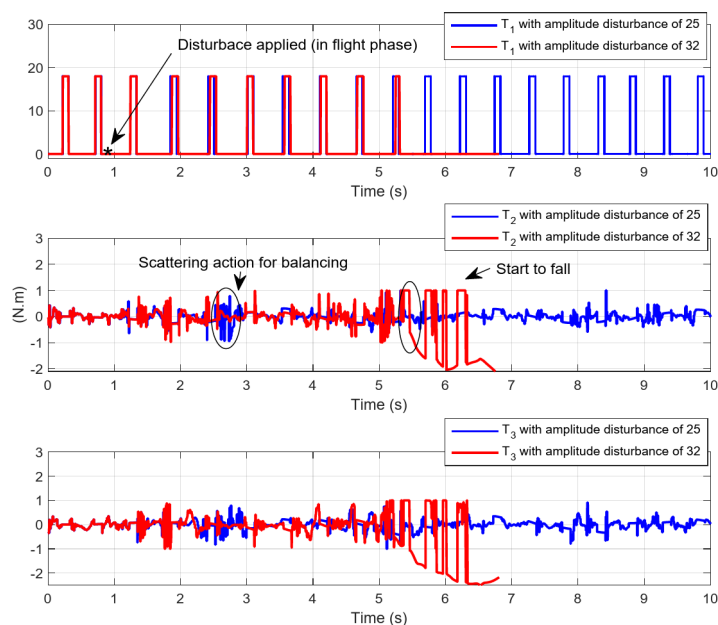


Fig. 16. Torques of actuators applied in presence of impulse disturbances.

Motion with horizontal impulse disturbances

This section presents findings concerning the assessment of the robot's performance subsequent to an impact directed along the horizontal axis toward its upper body. The results encompass two distinct simulations: the first involves an impact with an amplitude of 25 N and an impact duration of 0.1 s on the robot's upper body, while the second one involves an impact with an amplitude of 32 N over the same 0.1 s duration on the robot's upper body.

According to Fig. 16, the simulation at 0.8 s illustrates the actuators' response, exhibiting a scattered behavior aimed at compensating for the displacement effect. Consequently, the actuators repeatedly reach the torque saturation limit within a brief period. This behavior suggests that the agent, lacking prior exposure to such displacement during the learning phase, attempts various torque applications to navigate the unfamiliar positioning resulting from sudden upper body acceleration. However, upon mitigating the initial acceleration post-impact, the agent skillfully adjusts the joint movements of the robot, facilitating a controlled landing that is based on its current position, while disregarding the effects of the impact.

In the simulation involving a force range of 32 N, the impact's heightened intensity and the prolonged duration of erratic agent behavior led to an unsuccessful landing. Here, post-impact, the robot endeavors to adjust its leg angle when near the ground, lacking the opportunity to alter the angle as depicted in Fig. 17.

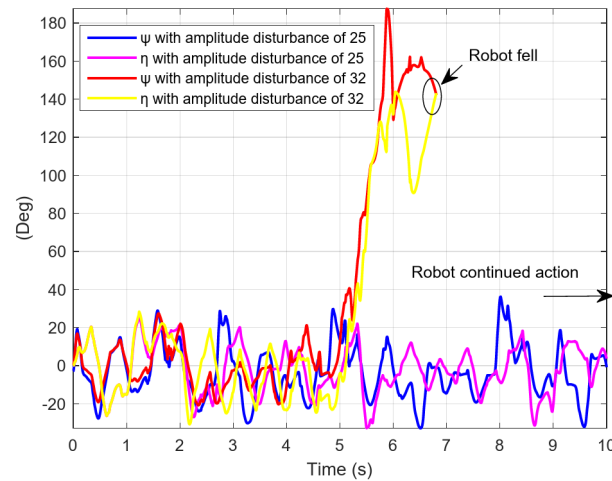


Fig. 17. The robot's foot angles in presence of impulse disturbances.

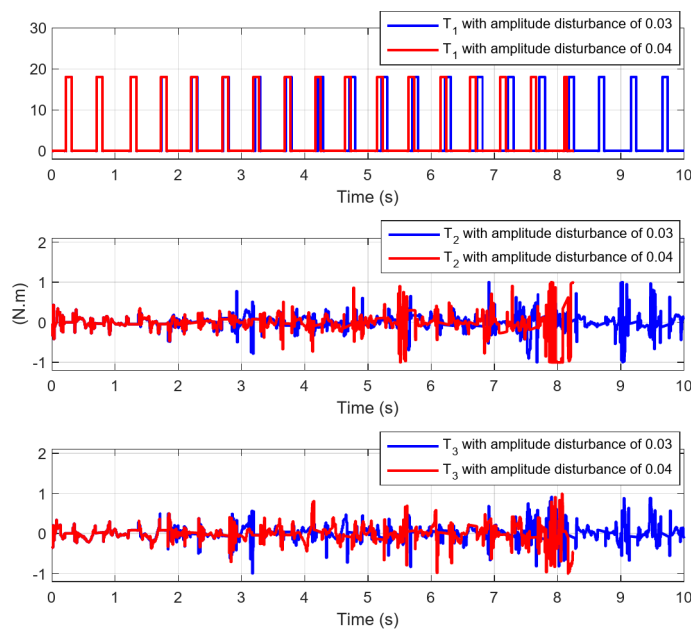


Fig. 18. Torques of actuators applied in presence of height change disturbances.

Movement on the surface with height difference

This section includes two simulations designed to evaluate the effects of changes in ground height displacement. In the initial test scenario, the robot makes contact with a surface positioned 3 cm lower than the preceding surfaces during the third step. Findings indicate the robot's capability to effectively manage this displacement, maintaining its equilibrium in subsequent steps. The simulation is structured to expose the robot to ground height discrepancies, occurring specifically during the third and thirteenth steps of impact with the ground, as depicted in Figs. 18 and 19.

However, when confronted with a 4 cm height difference (as illustrated in Fig. 20), the robot's equilibrium remains compromised, failing to sustain balance for more than four jumps after the second impact, (the 13th step), leading to its eventual fall.

Movement on a rough surface

In the third step, the robot faces a challenging terrain, as illustrated in Fig. 21. This challenge is represented by modifying the angle around the y-axis during this phase. In the first trial, the angle is adjusted by 3°, resulting in the robot interacting with a surface that is inclined at a 3° angle in relation to the horizontal plane. Findings indicate the robustness of the control mechanism against this displacement, as the robot adeptly maintains its balance for a duration of 10 s.

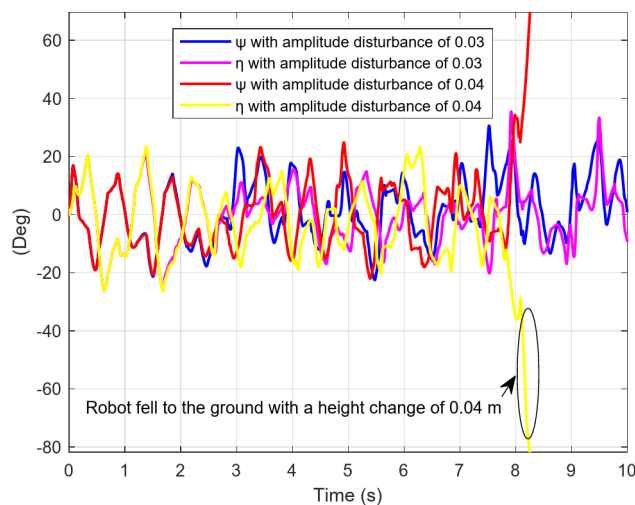


Fig. 19. The robot's foot angles in presence of height change disturbances.

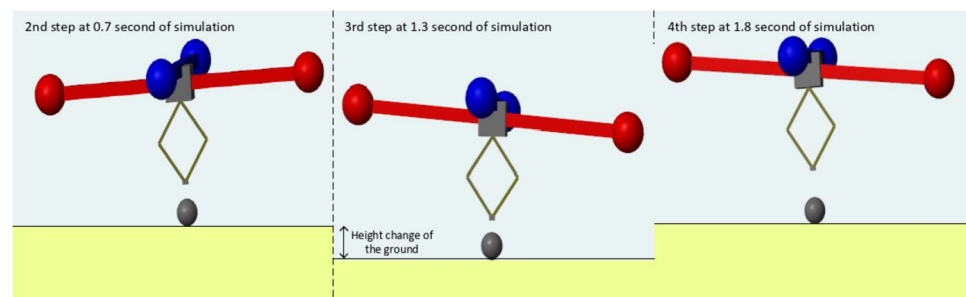


Fig. 20. The difference in the height of the ground in the third step of the jump.

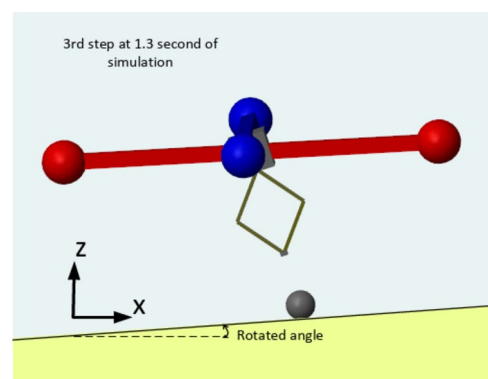


Fig. 21. Simulation on uneven ground.

As evidenced by Figs. 22 and 23, as the displacement range extends from 3° to 4°, the control system's efficacy in mitigating the impact of surface irregularities diminishes significantly, resulting in the robot's loss of stability and subsequent fall after approximately 7 s.

Conclusion

The study outlined in this paper showcased a model in Sect. 2 that demonstrated the ability to perform consecutive jumps through the use of a semi-active mechanism consisting of four links and a spring. Following this, Sect. 3 introduced a control mechanism that integrated feedback control and reinforcement learning. Finally, Sect. 4 detailed the results of multiple tests carried out on both the model and controller.

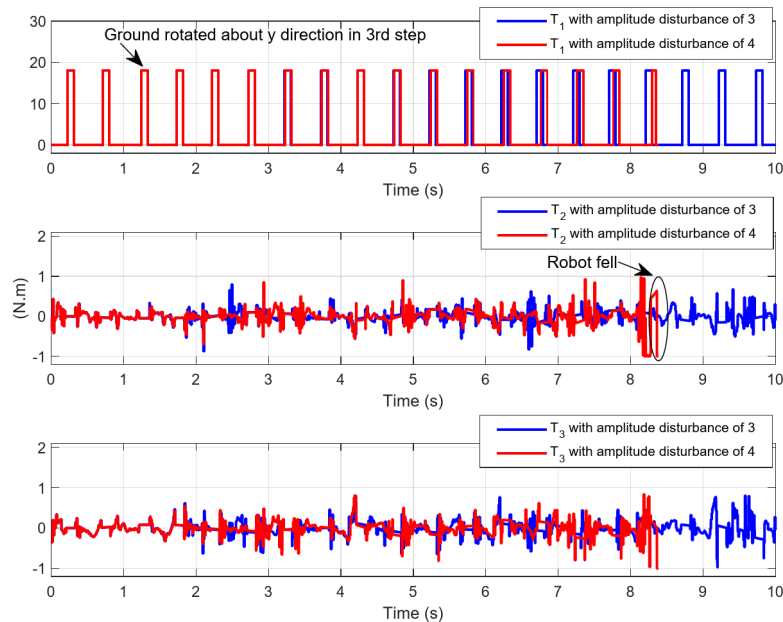


Fig. 22. Torques of actuators applied in presence of uneven surface disturbances.

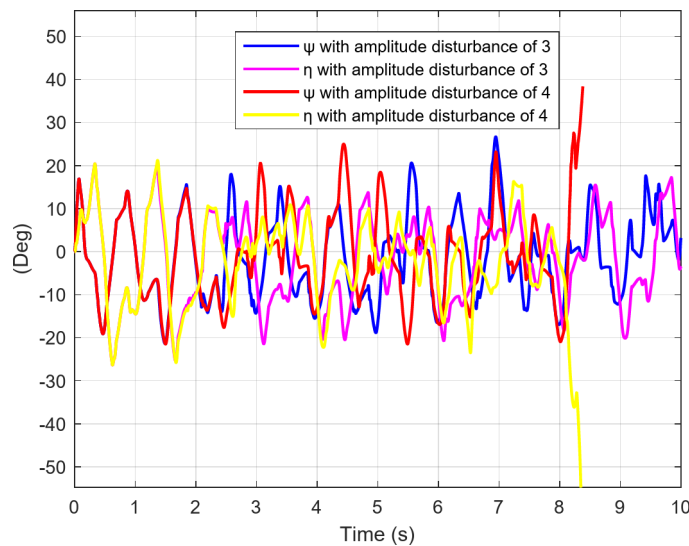


Fig. 23. The robot's foot angles in presence of uneven surface disturbances.

In this methodology, the reinforcement agent guides the robot in determining the suitable angle at which the foot should land when it is detached from the ground. This task is achieved by the agent through the utilization of torque generated by two robot actuators, which involves arm movements. Once the foot makes contact with the ground, the joint torque of the arms is controlled using PD controllers. Additionally, a multi-phase control method is employed to execute the jump maneuver.

According to the findings, the robot exhibits the capability to cover a distance of approximately 8 m on a flat surface and perform around 80 jumps within a time span of 40 s, all without encountering any falls. Moreover, the robot has demonstrated resilience against surface displacements and body impacts, being able to withstand impacts of up to 25 N. However, if the impacts exceed this threshold, the robot will fall after completing three jumps.

Moreover, in cases where the height variation between jumps exceeds 3 cm or the ground terrain displays irregularities surpassing 3°, the robot encounters difficulties in sustaining equilibrium after a limited number of jumps. The outcomes of the research indicate that the reward system is accurately established, as illustrated by the consistent increase in rewards per episode, showcasing the robot's capacity to acquire greater rewards through effective maneuvers.

The findings also suggest that the multiphase control feedback system for jumping successfully offsets the energy usage in every step, ensuring a consistent jump height of around 40 cm (equivalent to the robot's upper body height). Nevertheless, the absence of coordination between the torque applied, derived from height feedback and joint contraction, diminishes the robot's stability. The authors recommend additional research on robotic control mechanisms to improve the coordination between arm gestures and leg jumping actions.

Furthermore, the robot's strong reaction throughout the simulation is primarily attributed to the high sensitivity of the output torque. Consequently, this sensitivity, combined with the lightweight structure of the foot, leads to the robot's foot landing at incorrect angles, ultimately resulting in falls. To address this concern, it becomes crucial to develop a low-level controller that is responsible for applying torque to the joints. This controller should ensure that movement commands are exclusively issued by the AI agent, while the lower-level controller effectively executes these commands.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 15 June 2024; Accepted: 17 December 2024

Published online: 30 December 2024

References

1. Raibert, M. H. & Benjamin Brown, H. Jr Experiments in balance with a 3D one-legged hopping machine. *Int. J. Robot. Res.* **3** (2), 75–92 (1984).
2. Schwind, W. J., Daniel, E. & Koditschek Control of forward velocity for a simplified planar hopping robot. In *Proc. of 1995 IEEE International Conference on Robotics and Automation* vol. 1, pp. 691–696 (IEEE, 1995).
3. Fiorini, P. & Burdick, J. The development of hopping capabilities for small robots. *Auton. Robots.* **14**, 239–254 (2003).
4. Yu, H., Gao, H. & Deng, Z. Toward a unified approximate analytical representation for spatially running spring-loaded inverted pendulum model. *IEEE Trans. Robot.* **37** (2), 691–698 (2020).
5. Zhao, J. X. et al. One-legged hop of compliance control based on minimum-jerk. *J. Phys. Conf. Ser.* **1507**(5), 052012 (2020).
6. Iida, F. Exploiting friction for the locomotion of a hopping robot. In *Proc. of the 2nd Int. Symp. on Adaptive Motion of Animals and Machines* (2003).
7. Zhao, J., Zhong, J. and Fan, J. Position control of a pneumatic muscle actuator using RBF neural network tuned PID controller. *Mathematical Problems in Engineering* **2015** (2015).
8. Han, B., Luo, X., Liu, Q., Zhou, B. & Chen, X. Hybrid control for SLIP-based robots running on unknown rough terrain. *Robotica* **32**(7), 1065–1080 (2014).
9. Han, B., Yi, H., Xu, Z., Yang, X. & Luo, X. 3D-SLIP model based dynamic stability strategy for legged robots with impact disturbance rejection. *Sci. Rep.* **12** (1), 5892 (2022).
10. Harbick, K. & Gaurav, S. Sukhatme. Height control for a one-legged hopping robot using a two-dimensional model. In Tech. Rep. IRIS-OI-406. Institute for Robotics and Intelligent Systems, University of Southern California (2001).
11. Abe, Y. & Katsura, S. Compensation of nonlinear dynamics for energy/phase control of hopping robot. *Precis. Eng.* **69**, 36–47 (2021).
12. Wang, Y., Zhu, Q., Xiong, R. & Chu, J. Standing balance control for position control-based humanoid robot. *IFAC Proc.* **46**(20), 429–436 (2013).
13. Scianca, N., Simone, D. D., Lanari, L. & Oriolo, G. MPC for humanoid gait generation: Stability and feasibility. *IEEE Trans. Robot.* **36** (4), 1171–1188 (2020).
14. Samadi, S., Caron, S., Tanguy, A. & Kheddar, A. Balance of humanoid robots in a mix of fixed and sliding multi-contact scenarios. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6590–6596 (IEEE, 2020).
15. Morita, Y. & Ohnishi, K. Attitude control of hopping robot using angular momentum. In *IEEE International Conference on Industrial Technology* **1**, 173–178 (IEEE, 2003).
16. Razzaghi, P., Al Khatib, E. and Hurmuzlu, Y. Nonlinear dynamics and control of an inertially actuated jumper robot. *Nonlinear Dyn.* **97**, 161–176 (2019).
17. He, G. & Geng, Z. Robust backstepping control of an underactuated one-legged hopping robot in stance phase. *Robotica* **28**(4), 583–596 (2010).
18. Goswami, D. & Vadakkepat, P. Planar bipedal jumping gaits with stable landing. *IEEE Trans. Robot.* **25** (5), 1030–1046 (2009).
19. Kashki, M., Zoghzy, J. & Hurmuzlu, Y. Adaptive control of inertially actuated bouncing robot. *IEEE/ASME Trans. Mechatron.* **22** (5), 2196–2207 (2017).
20. Shemer, N. & Degani, A. A flight-phase terrain following control strategy for stable and robust hopping of a one-legged robot under large terrain variations. *Bioinspir. Biomim.* **12** (4), 046011 (2017).
21. Calderón, J. M., Moreno, W. & Weitzenfeld, A. Fuzzy variable stiffness in landing phase for jumping robot. In *Innovations in Bio-Inspired Computing and Applications: Proceedings of the 6th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2015) held in Kochi, India during December 16–18*, pp. 511–522 (Springer International Publishing, 2016).
22. Khadiv, Majid, S., Ali, A. & Moosavian Aghil Yousefi-Koma, Hessam Maleki, and Majid Sadedel. Online adaptation for humanoids walking on uncertain surfaces. In *Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* **231**(4), 245–258 (2017).
23. Khadiv, M. S. et al. Rigid vs compliant contact: an experimental study on biped walking. *Multibody Syst. Dyn.* **45**, 379–401 (2019).
24. Qu, D., Zheng, Y., Guo, J. & Song, R. A control scheme for single legged hopping robot based on fuzzy PD algorithm. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* **1**, pp. 384–388 (IEEE, 2020).
25. Yim, J. K., Singh, B. R. P., Wang, E. K., Featherstone, R. & Ronald, S.F. Precision robotic leaping and landing using stance-phase balance. *IEEE Rob. Autom. Lett.* **5** (2), 3422–3429 (2020).
26. Ugurlu, B., Sariyildiz, E. & Kawasaki, T. Agile and stable running locomotion control for an untethered and one-legged hopping robot. *Auton. Robots* **45**, 805–819 (2021).
27. Tutsoy, O., Barkana, D. E. & Colak, S. Learning to balance an NAO robot using reinforcement learning with symbolic inverse kinematic. *Trans. Inst. Meas. Control* **39** (11), 1735–1748 (2017).
28. Yi, S. J., et al. Learning full body push recovery control for small humanoid robots. In *2011 IEEE International Conference on Robotics and Automation*, pp. 2047–2052 (IEEE, 2011).
29. Wang, S., Braaksma, J., Babuska, R. & Hobbelen, D. Reinforcement learning control for biped robot walking on uneven surfaces. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 4173–4178 (IEEE, 2006).
30. Hwangbo, J. et al. Learning agile and dynamic motor skills for legged robots. *Sci. Rob.* **4**, 26 (2019).

31. Hoseinifard, S., Mohamad & Sadedel, M. Analysis of stability and horizontal motion of a single leg hopping robot. *J. Intell. Robot. Syst.* **108** (4), 73 (2023).

Author contributions

S.Mohamad Hoseinifard: Methodology, investigation, writing, analyzing, visualization, software. Majid Sadedel: Supervision, project administration, conceptualization.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-024-83749-x>.

Correspondence and requests for materials should be addressed to M.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024