2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Self-rising Bipedal Robot for Embracing Fall Impact and Fall Detection with Multimodal Sensing

Kenta Hirashima[1], Daniel Campos Zamora[2], Kevin G. Gim[1], and Joohyung Kim[1]

***Abstract*— Humanoid robots are inherently unstable, making fall management critical, especially for hardware-based reinforcement learning (RL), where falls frequently occur. This paper introduces a lantern-shaped mechanical cover designed for a kid-sized humanoid robot to mitigate damage during falls and support autonomous recovery. A multimodal fall detection method integrating inertial, proprioceptive, and acoustic sensors was implemented alongside an improved stance phase detection algorithm that eliminates reliance on heuristic thresholds. Hardware experiments on the Hybrid Leg biped robot demonstrated improved walking robustness and revealed a 57.4% success rate for autonomous recovery after induced falls. Results indicated that perturbations in vertical (z) and positive forward (x) foot trajectories posed the greatest challenges to successful recovery.**

I. INTRODUCTION

Humanoid robots are often more flexible to adapt to different environments than wheeled or multi-legged robots. However, their stable and reliable locomotion remains a challenge. Recently, learning-based methods have drawn much attention as a potential solution to address the challenges in humanoid locomotion [1]–[5]. While the results are stunning, the learning-based methods typically require a large number of trials, often in the order of magnitude of thousands [6]. Another challenge is the *sim-to-real* gap, which can cause performance degradation of a policy trained in a simulation environment.

One workaround for this issue is the direct application of the learning-based methods to a real robotic system and automating the learning process. However, this approach comes at the cost of requiring re-initialization, which is standing back up and restarting the training after each episode. For some lightweight robots, an external resetting device was used for re-initialization, such as a 1 Degree of Freedom (DoF) system or a robotic arm [7], [8]. Kohl and Stone explored an automated environment for quadrupedal learning [9]; however, it required a reasonable starting policy that prevents falls for an effective learning process. Moreover, the inherent instability of the humanoid makes it challenging to completely avoid falls with an initial policy and often results in human intervention for a manual reset between roll-outs. However, human interventions during the process negatively affect the scalability of the learned policy. Thus, it is also crucial to minimize the human intervention [10].

[1]Kenta Hirashima, Kevin G. Gim, and Joohyung Kim are with the KIMLAB (Kinetic Intelligent Machine Lab), University of Illinois Urbana-Champaign, IL 61801, USA. [2]Daniel Campos Zamora is with University of Washington, Seattle, Seattle, WA 98195, USA. Author contact information is {kentah2, joohyung}@illinois.edu

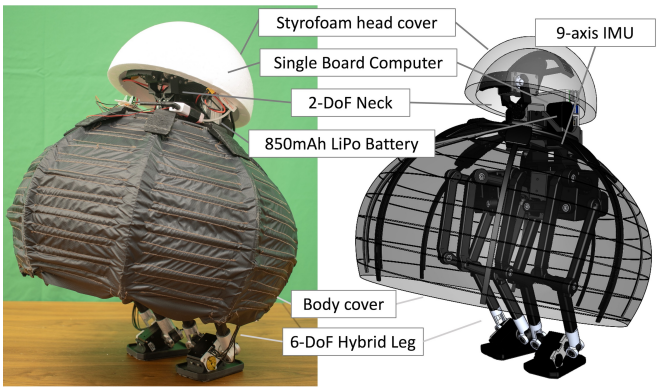


Fig. 1. Hybrid Leg bipedal system with body cover and Styrofoam cover for head protection

Therefore, for applications such as hardware RL, it is crucial to develop a system capable of autonomous re-initialization, where a key component for unstable platforms like humanoid robots is fall management—comprising both fall detection and damage mitigation.

There are several different approaches to detecting a fall on a humanoid robot. For example, Renner *et al.* compare the expected readings from attitude sensors in normal gait to the actual sensor readings to trigger an instability detector [11]. Another approach uses the Capture Point (CP) based on 3D LIPM dynamics and detects a fall when it does not lie within the support polygon [12]. Moreover, Höhn *et al.* used feature vectors to classify patterns and detect a fall. Meanwhile, Ruiz-del-Solar *et al*. made a more direct usage of attitude sensors to detect falls along with a threshold built online [13].

On the topic of fall mitigation, Fujiwara *et al.* explored a unique strategy inspired by UKEMI motion in Judo, proposing a method to control the landing so that impact occurs at a specific limb or body component [14]. Along the same lines, Ogata *et al.* introduced some shock-reducing motions for falls in the sagittal plane [15]. These strategies, like UKEMI, accept that some falls are unavoidable and instead focus on minimizing damage. A comparable philosophy was adopted in [16], where the design accommodates falls rather than strictly avoiding them. In this study, we adopt a damage-tolerant design philosophy, implementing a mechanical solution that enables the robot, despite the absence of arms, to withstand impacts and autonomously recover from falls. In our previous work [17], we implemented an untethered bipedal platform that uses a leg structure called *Hybrid Leg*

978-1-6654-5784-2/26/$31.00 ©2026 IEEE

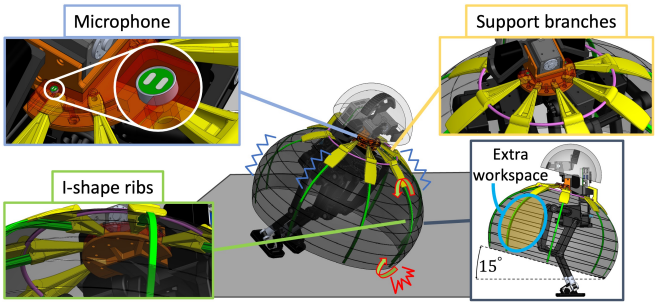


Fig. 2. Hardware overview of the covered Hybrid Leg bipedal platform

[18], [19]. We made the walking controller based on the ZMP Preview Control [20] for the robot and performed walking experiments. In this study, we upgraded the hardware with a cover structure of lantern-shape. Moreover, a fall is detected using a combination of sensor readings. In addition to IMU and proprioceptive sensors, microphones are used as contact sensors, similar to the usage in [21]. Instead of directly detecting fall contact with a contact sensor which bears the risk of sensor damage, microphones can provide an indirect and more robust way to detect a fall. The recovery motion is taken afterward to return the robot to a stand-up pose. Additionally, stance phase detection for the walking controller was modified to eliminate the dependency on a heuristically determined threshold.

The rest of this paper is organized as follows. In Section II, we detail the hardware modifications and improvements from the previous bipedal platform, with a focus on the cover structure and microphone. In Section III, we present the newly implemented method of stance phase detection, as well as the fall detection scheme and fall recovery motion. In Section IV, the walking performance between the previous and new stance phase detection algorithm is tested. Further, the fall recovery and its success rate is evaluated. Finally, in Section V, we discuss conclusions and future works.

II. Hardware Implementation

A. Hybrid Leg Bipedal System

Aside from the cover and addition of the microphone as well as modified neck design, most of the bipedal system is the same as the implementation shown in [17]. The entire system is powered by a voltage supply from two 850 mAh 3-cell Lithium Polymer (LiPo) batteries with a 75 C rating.

B. Body Cover and Cover Support Structure

The cover structure holds the cover at a 15-degree angle with respect to the horizontal, as shown in Fig. 2. The primary goal of this cover installation is to protect the robot and its legs when falling. The leg structure is one of the most susceptible components at impact as it is often the first structure to touch the ground. Thus, the cover aims to contain the impact and maximally mitigate the damage to the legs. However, a horizontally attached cover interferes with some leg links. Therefore, the cover is oriented 15 degrees upward to take full advantage of the leg workspace while protecting the robot. For the cover structure, as opposed to a square shape like the one presented in [16], the round shape was chosen to maximize the workspace inside the cover while minimizing the overall size. Additionally, it was designed with a creative intent, similar to the design philosophy adopted in the work by Grandia *et al.* [5] .

The outer skin of the body cover is made of ripstop-patterned nylon fabric. Fabric pieces are cut into gore segments and sewn together with topstitching to form a lantern-like shape. For reinforcement, carbon fiber strips run horizontally on the inner side, acting like hoops on a lantern. Most of the structural load is held up by the ribs and hoops, but the cover serves as an alternative medium for wave propagation as well as an aesthetic component. The cover attaches to the robot via vertical ribs and a top ring where the gore segments converge. As shown in Fig.2, eight branch-like structures extend from the robot's neck. The front five branches snap onto the top ring, while the back three have the ring pass through a hole for better attachment. The front branches also have slots underneath for inserting ribs. The cover must support the robot's weight and absorb impact in a fall, so the ribs have an I-beam cross-section for improved bending resistance. The I-shape tapers at the bottom to fit into pockets in the skin.

Since the neck parts are 3D printed with PLA, a rigid connection between the cover support structure and neck could lead to fractures after repeated impacts. Although the ribs and hoops on the cover inherently dampen impacts, some energy can still propagate to the neck. Thus, to make the system more robust, the branch structures connect to the central disk on the neck, as shown in the top-right of Fig. 2. The branch tip geometry prevents detachment while allowing slight movement to further dampen impacts.

An additional minor modification from the previous Hybrid Leg platform is the repositioning of the battery. Since the cover encapsulates the robot's body structure, accessing the battery previously housed within it was difficult. To improve accessibility, the batteries were relocated above the opening of the cover.

C. Microphone as Contact Sensor

For fall detection, while acceleration measurement is simpler, differentiating that of gait and fall was unreliable. Therefore, we utilize a small electret capacitor microphone. Two of these microphones are placed on the central disk with the sound port facing into the structure. Microphones can be excited either by mechanical vibration or acoustic pressure field [22]. To take full advantage of the microphone's capability, the sound port is exposed to the opening where the tip of the branch resides (see top-left of Fig. 2). When the robot falls and the cover impacts the ground, the entire robot structure undergoes some vibration. The embedded placement is intended to maximize the input to the microphones when this vibration occurs.

Two microphones are wired to a single TRS connector, which connects to a sound card module, serially communi-

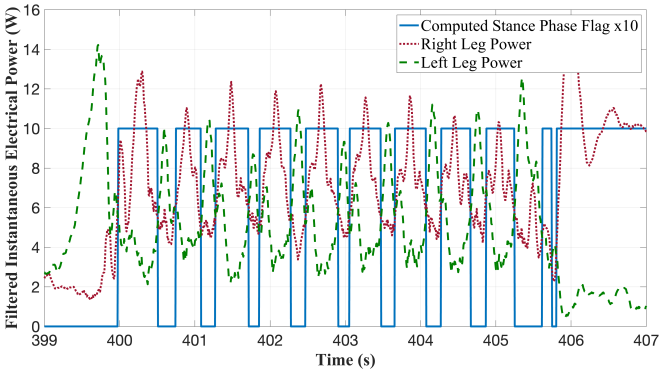


Fig. 3. Stance phase detection: stance phase flag of 10 indicates right, and 0 indicates left.

cating with the SBC. The sound card used in the system can only handle mono sound input, so the two microphones effectively serve as two different sources of mono sound input.

III. Software Implementation

A. Stance/Swing leg detection

While phase information could be provided by the walking controller, the leg actuation may not exactly follow the desired phase timing. As such, for a better stabilization control, stance phase is an important state to estimate for a humanoid robot. A common approach is to equip the foot with force-sensing capability with a force/torque sensor or load cell. Some unique approaches use an array of skin sensors and utilize their haptic feedback [23]. Since our bipedal platform does not have any sensor equipped at the foot sole, filtered current measurement from the ankle motors was used for detecting stance phase switching in the previous implementation. However, the current measurement from motors often exhibits noisy spikes. More importantly, a heuristically determined current threshold likely becomes invalid if the operating condition is drastically changed, or if the hardware is largely modified.

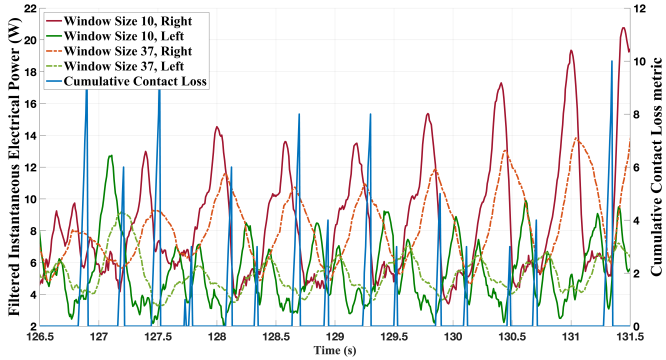
Thus, for the new implementation, we compute a simple metric for each leg and use the comparison between these metrics for stance phase detection. The metric is a product of measured voltage and measured current, which effectively gives an approximation of instantaneous electrical power supplied to the motor. While the Hybrid Leg structure distributes some of the foot's DoF across four pitch motors at the pelvis, the remaining DoF is actuated by the pelvis roll motor and the ankle pitch motor. Consequently, these two actuators are subject to the highest loads during single stance phase. To reflect the overall loading condition of a leg, we compute a metric by summing the instantaneous power of the ankle pitch and pelvis roll motors. As this metric can still be noisy, it was filtered with a moving average filter of size 10 for a topic publishing rate of 125 Hz. A plot showing the filtered metric for both legs in 20-step forward walking is shown in Fig. 3. The same plot also shows the computed contact phase, which is calculated by a simple comparison of the metric for the left leg and right leg. The algorithm determines that if the right metric is larger than the left one, the robot is in the right stance phase, and otherwise, the robot is in the left stance phase. This method avoids the usage of a threshold for stance phase detection, which reduces the number of variables necessary for walking controller tuning. One of the limitations of this method is the inability to distinguish double support phase (DSP) from single support phase (SSP), which leads to some inaccuracy in detecting the real contact switch timing. The unequal stance duration between left and right legs, shown in Fig. 3, is mainly caused by the gait adjustment from the walking controller, but the aforementioned inaccuracy is another contributing factor.

B. Fall Detection Algorithm

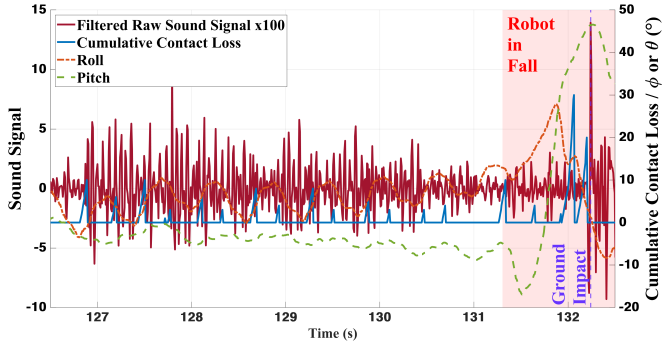
Fall management can be broadly categorized into four primary aspects: robot design, instability detection, fall avoidance, and damage mitigation or recovery [13]. In this paper, our main focus is on the integrated robot design as part of the damage mitigation, as well as fall recovery after detecting a fall. The model-based approaches to fall detection rely on good estimation of velocity or acceleration of Center of Mass (CoM) for calculation of metrics like CP or ZMP. However, for our system, because of the lack of foot sensors, it is challenging to estimate these quantities well. Thus, we take a simpler approach and combine several sensor measurements to detect a fall. Namely, we use the measurements from proprioceptive sensors in the actuators, IMU, and the microphones.

1) Algorithm flow: We employ the algorithm in the form of a hierarchical state machine. It keeps two different levels of alert state, and at last, the state transitions to a fall state. The state transitioning is based on measurements from the sensors mentioned above and corresponding thresholds. Initially, the first thresholding of IMU orientation with the threshold of 10 degrees raises the state from nominal to first alert state. After holding the first alert state for at most 0.3 seconds, if the microphone picks up a noise greater than the microphone threshold, the state is raised to the next level, which lasts at most 0.4 seconds. At last, there are two ways in which the state could shift to the fall state. One way is if the cumulative contact loss (CCL) metric takes a value greater than or equal to 10, and the foot contact loss is detected. Otherwise, the fall state can also be triggered by satisfying a higher IMU orientation threshold of 13 degrees and a lower CCL threshold of 2. The second condition was added to detect a fall as early as possible, as a delayed reaction to a fall often led to a non-recoverable fall. The alert state durations and thresholds were chosen heuristically.

2) Foot contact loss detection: For foot contact loss detection, we use the same metric used for contact switch detection. However, in this case, we keep moving averages of two different window sizes: one window with a size of 10 and another with a size of 37. When the robot falls, and the foot contact is lost, the value of the metric decreases. To detect this decrease without thresholding the metric itself, we compare the values of two moving averages. When there



(a) Contact Loss Detection with Motor Power



(b) Fall Detection with all the metrics

Fig. 4. Plots showing the values of various metrics used for fall detection

is a sudden change in the metric, the moving average with a larger window size lags the one with a smaller window. In nominal walking gait, this occurs only for either the left or right leg at a time. Thus, we condition the contact loss detection to trigger when the lag of a larger window moving average occurs for both legs simultaneously. This use of two moving averages is similar to moving average crossover technique used in stock market technical analysis. One caveat is that, with a simple conditioning of larger moving average window lagging for both legs, contact loss is falsely detected during the DSP as well. To bypass this issue, the algorithm calculates a CCL metric, which exponentially increases the longer the contact loss is detected continuously. It is based on the current binary output of the contact loss detection function CL_{curr} and the output from the control loop immediately before CL_{prev}. The CCL is defined as follows:

$$CCL = ((CL_{prev} + CL_{curr}) * CL_{curr})^{1.05}. \quad (1)$$

To deem the foot contact to be lost, we apply a threshold to this CCL metric instead of the instantaneous electrical power. Keeping in mind that the contact loss can falsely be detected during DSP and thus increase the CCL, the threshold should at least be as large as the value it takes after the duration of DSP to avoid a false positive. Since the duration of DSP of the chosen gait is 0.06 s and the control loop is 8 ms, starting from initial $CL_{prev} = 0$, after seven consecutive updates with $CL_{curr} = 1$, the CCL can increase up to slightly above 9. Thus, the threshold for CCL is chosen to be 10. Fig. 4a shows the moving averages of two different sizes for both legs and the computed CCL.

3) Microphone for contact detection: For microphone streaming, we mounted two electret microphones near the robot's neck. We handled audio input using ALSA, Linux's native sound interface. Because the raw microphone signal saturated without attenuation, we reduced its amplitude before sampling. Although ALSA operates at 44.1 kHz, the Raspberry Pi ROS node could not process or transfer data at that rate. Therefore, we down-sampled the audio by averaging 8 ms windows. This introduces aliasing, but most relevant noise energy lies below 40 Hz—well under the resulting 62.5 Hz effective sampling rate. We also applied a 500 Hz low-pass filter to smooth the signal.

A simple thresholding technique was used for sound-based signal triggering. However, microphones were activated not only by fall impact but also by vibration from nominal walking. While setting a high threshold may mitigate false positives, it is more critical not to miss a fall than to avoid detecting a false one. Thus, we chose to set a low threshold that assures the detection of any fall and tolerated false positives for the sound-based triggering of the second alert state. The overall false positive of fall detection is mitigated by triggering the fall state based on a metric that is less susceptible to false positives.

Fig. 4b shows the sound signal plot along with the roll and pitch measurements from IMU, as well as the CCL. It is the same set of data as Fig. 4a. For the purpose of plotting, the raw signal was scaled by 100 times. The red-shaded region starts when the fall is detected. The robot starts to become unstable just around when the fall is detected, and then the robot tips over and hits the ground, which is indicated by the peak in pitch measurement that coincides with the large spike of the sound signal. While the most reliable indication of a fall is ground impact, the compliant nature and the shape of the cover necessitated an earlier detection of fall for recovery motion to activate on time, ultimately preventing the robot from completely rolling over. For achieving earlier detection, the threshold on the microphone signal was adjusted to a low value of 0.0001, or 0.01 in the scaled plot. Table I shows the number of true positives (TP), false positives (FP), and false negatives (FN) for fall detection using each individual metric, as well as the proposed algorithm that combines multiple metrics. The experiment was conducted with 117 falls. By utilizing each metric in a hierarchical manner, the proposed detection method maintains a high fall detection rate while reducing false positives.

C. Fall Recovery Motion

The fall recovery motion refers to the entire sequence, starting from detecting a fall, embracing impact, and standing back up. The sequence of motions employed in this study can largely be broken into four stages: Fall detection, Leg extension, Leg retraction, and Standing back up.

The sequence of leg motion is also illustrated in Fig. 5. One of the features of the Hybrid Leg design is that most of

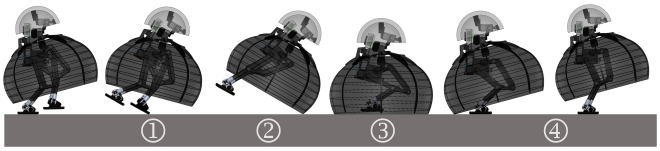


Fig. 5. Recovery motion sequence: (1) fall detection, (2) leg extension, (3) leg retraction, and (4) standing back up

TABLE I. Comparison of Detection Results Across Methods

Method	TP	FP	FN
IMU only	117	73	0
CCL only	90	4	27
Microphone only	117	112	0
Proposed Method	117	4	0

its actuators are concentrated at the pelvis. This design makes it suitable for applying the walking pattern generation based on LIPM or cart-mass table model, such as the ZMP Preview control [20]. However, in case of falling, combined with the round shape of the cover, the concentrated CoM makes the robot prone to rolling over completely, resting on its head at the end. To offset the CoM, immediately after the fall gets detected, legs are extended as fast and far as possible. Another aim of this rapid extension is to give the robot enough momentum to prevent it from a complete rollover.

The next motion is the leg retraction. After the leg extension phase has ended, the robot crouches and retracts its legs. At the end of this phase, the legs are completely concealed inside the cover, which also supports the robot's weight.

The last motion is standing back up. At the beginning of this phase, the foot is aligned parallel to the ground and is moved to an appropriate position so that the CoM of the robot rests above the support polygon. The legs are then slowly extended while avoiding tipping over. When the robot is standing on its feet with a nominal CoM height from the ground, the fall recovery is complete.

IV. Bipedal Walking and Fall Recovery Test

A. Walking Performance Comparison

First, a simple walking performance comparison between two different implementations of stance phase detection was conducted. A 20-step forward walking command was given to the robot, and the number of times that the robot successfully completed the steps without falling was recorded. For the duration of the experiment, LiPo batteries were replaced when their voltage reached about 11.0 V. 110 walking trials were conducted for each stance phase detection implementation, and the results are shown in Table II.

The percentage of falls experienced decreased by about 6% for the new stance phase detection method, which implies a comparable walking performance even without reliance on a heuristically determined current threshold. However, devising a method for managing falls and facilitating fall recovery is still important.

TABLE II. Comparison of Walking Performance with Two Stance Phase Detection Methods

Detection Method	No Fall	Fall	Total
Current Threshold Based	59	51	110
Instantaneous Electrical Power Based	65	45	110

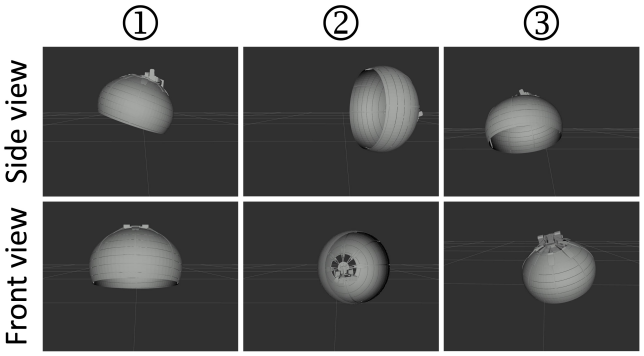


Fig. 6. Rviz cover visualization at different states: (1) nominal, (2) non-recoverable fall, and (3) recoverable fall

B. Fall Recovery Test

A test was performed to collect data on the rate of successful fall recovery. For each trial, a forward walking motion of the robot is initiated by a joystick command from a Steam Deck, and after 7 or 8 steps, a perturbation was given to a foot position command. There are six possible foot position commands that can receive a perturbation; they are the left or right x , y, and z position of the foot trajectory command, expressed with respect to its local coordinate frame where x points forwards, y points to the left, and z points up. The perturbation was given in the form of an offset to the nominal trajectory calculated by a foot trajectory server node, and the magnitude of perturbation was randomly chosen from the range of 10 to 40 mm, and its sign is also randomized. A constant perturbation is given for the duration of a step time, which is 0.3 seconds.

A total of 204 trials were performed, and the end state of the robot was classified as either recovered, non-recoverable, or no fall. The recovered state indicates that the robot was able to detect a fall in time and rest on its cover to stand back up. A non-recoverable state occurs when the robot is able to detect a fall but falls in a manner that ultimately rests on its head, unable to reposition itself with its feet down on the ground. Finally, no fall indicates that the perturbation was insufficient to cause a fall. Fig. 6 shows Rviz visualization of the cover at different states. It estimates the cover state based on kinematic information and IMU data, which helps approximate the location of impact on the cover.

The results are as shown in Table III. The percentage is calculated by dividing the number of occurrences of each case by 204, which is the total number of trials. Excluding the "no fall" case, 42.6% of the falls ended in a non-recoverable state, while 57.4% resulted in successful fall recovery. The z-perturbation has the highest fatality rate

TABLE III. End State Classification by Perturbation Direction and Sign (%)

Direction	Sign	Recovered	Non-recoverable	No Fall
x Perturbation	+	7.35	11.27	2.45
	−	13.73	0.98	2.94
y Perturbation	+	7.35	2.94	4.90
	−	8.82	2.94	1.96
z Perturbation	+	6.37	7.84	0.98
	−	5.88	10.78	0.49
Total		**49.51**	**36.76**	**13.73**

of 57.5%, which is defined by the total number of non-recoverable falls from z-perturbation divided by the total number of z-perturbations given. It is followed by 31.6% for x and 20.3% for y-perturbations. The perturbations in the z-position of a foot most likely result in a sideways fall, and the rounded shape of the cover makes it challenging to react in time or recover from a sideways fall. Therefore, the z-perturbation is the most fatal to the robot. Furthermore, of the x-perturbations given, relative to the negative perturbation, the positive perturbation was about 10 times more likely to result in a non-recoverable fall. The y-perturbations overall, on the other hand, are not as critical to the robot.

V. Conclusions and Future Work

This work upgraded the bipedal Hybrid Leg robot with a lantern-shaped cover to absorb impact during falls and support recovery. A new stance phase detection method, free from heuristic thresholds, slightly improved walking performance. Additionally, a fall detection system using three sensors was implemented. Recovery tests showed that foot trajectory perturbations, especially in the z and positive x directions, most hinder successful fall recovery.

For our future work, we plan to incorporate reinforcement learning into the covered Hybrid Leg bipedal platform. With the ability to recover from falls, the robot can now learn to walk more robustly through repeated trial-and-error. Moreover, there remains room for improvement in the overall walking performance and the rate of successful recovery. While a 57.4% recovery rate is not low, it should ideally be close to 100%. In addition, the 3D-printed parts in the cover support structure and the cover itself sustained damage after multiple falls, indicating that their durability could be further enhanced.

References

[1] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. Riedmiller, and D. Silver, "Emergence of locomotion behaviours in rich environments," 2017. [Online]. Available: https://arxiv.org/abs/1707.02286

[2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[3] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi9579, 2024.

[4] C. Yang, K. Yuan, S. Heng, T. Komura, and Z. Li, "Learning natural locomotion behaviors for humanoid robots using human bias," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2610–2617, 2020.

[5] R. Grandia, E. Knoop, M. Hopkins, G. Wiedebach, J. Bishop, S. Pickles, D. Müller, and M. Bächer, "Design and control of a bipedal robotic character," in *Robotics: Science and Systems XX*, ser. RSS2024. Robotics: Science and Systems Foundation, Jul. 2024. [Online]. Available: http://dx.doi.org/10.15607/RSS.2024.XX.103

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[7] S. Ha, J. Kim, and K. Yamane, "Automated deep reinforcement learning environment for hardware of a modular legged robot," in *2018 15th International Conference on Ubiquitous Robots (UR)*, 2018, pp. 348–354.

[8] K. S. Luck, J. Campbell, M. A. Jansen, D. M. Aukes, and H. B. Amor, "From the lab to the desert: Fast prototyping and learning of robot locomotion," 2017. [Online]. Available: https://arxiv.org/abs/1706.01977

[9] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, 2004, pp. 2619–2624 Vol.3.

[10] S. Ha, P. Xu, Z. Tan, S. Levine, and J. Tan, "Learning to walk in the real world with minimal human effort," 2020. [Online]. Available: https://arxiv.org/abs/2002.08550

[11] R. Renner and S. Behnke, "Instability detection and fall avoidance for a humanoid using attitude sensors and reflexes," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2967–2973.

[12] X. Xinjilefu, S. Feng, and C. G. Atkeson, "Center of mass estimator for humanoids and its application in modelling error compensation, fall detection and prevention," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 67–73.

[13] J. Ruiz-del Solar, J. Moya, and I. Parra-Tsunekawa, "Fall detection and management in biped humanoid robots," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 3323–3328.

[14] K. Fujiwara, F. Kanehiro, S. Kajita, K. Kaneko, K. Yokoi, and H. Hirukawa, "Ukemi: falling motion control to minimize damage to biped humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2002, pp. 2521–2526 vol.3.

[15] K. Ogata, K. Terada, and Y. Kuniyoshi, "Real-time selection and generation of fall damage reduction actions for humanoid robots," in *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 233–238.

[16] K.-N.-K. Nguyen, Y. Kojio, S. Noda, F. Sugai, K. Kojima, Y. Kakiuchi, K. Okada, and M. Inaba, "Dynamic fall recovery motion generation on biped robot with shell protector," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6741–6748, 2021.

[17] K. Hirashima, N. Myers, D. C. Zamora, K. G. Gim, and J. Kim, "Implementation of untethered biped robots utilizing serial-parallel hybrid leg mechanisms," in *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*, 2024, pp. 227–233.

[18] K. G. Gim, J. Kim, and K. Yamane, "Design of a serial-parallel hybrid leg for a humanoid robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6076–6081.

[19] ——, "Design and fabrication of a bipedal robot using serial-parallel hybrid leg mechanism," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5095–5100.

[20] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE International Conference on Robotics and Automation*, vol. 2, 2003, pp. 1620–1626 vol.2.

[21] X. Fan, D. Lee, Y. Chen, C. Prepscius, V. Isler, L. Jackel, H. S. Seung, and D. Lee, "Acoustic collision detection and localization for robot manipulators," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9529–9536.

[22] C. B. King and C. Monti, "Microphone vibration sensitivity: what it is, why it is important, and how to measure it," *Proceedings of Meetings on Acoustics*, vol. 50, no. 1, p. 065001, 02 2023. [Online]. Available: https://doi.org/10.1121/2.0001702

[23] J. Rogelio Guadarrama-Olvera, F. Bergner, E. Dean, and G. Cheng, "Enhancing biped locomotion on unknown terrain using tactile feedback," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 1–9.