Real-Time Hybrid Detection of Adversarial Attacks on Learning-Based Locomotion Controllers for Quadrupedal Robots

Ali A. Noghabi *
Amirkabir University of Technology
a.abdollahian@aut.ac.ir

Upinder Kaur †
Purdue University
kauru@purdue.edu

Abstract

Quadrupedal robots with learning-based locomotion controllers are highly adaptable but vulnerable to adversarial attacks that induce small perturbations in sensor data or control signals, potentially leading to catastrophic failures. This paper introduces a hybrid detection framework, combining a Kalman Filter and a Physics-Informed Neural Network (PINN), to detect adversarial attacks in real-time. Using a comprehensive dataset collected under various conditions, including normal and adversarial scenarios, we demonstrate significant improvements in detection accuracy and robustness.

1 Introduction

Learning-based locomotion controllers have shown significant potential in enabling quadrupedal robots to perform a wide range of dynamic tasks, such as walking, running, and maneuvering on uneven terrain. However, their vulnerability to adversarial attacks—small perturbations in sensor inputs or control signals that can drastically affect performance—poses a critical challenge.(1)

Previous works (2; 3; 4) have demonstrated how adversarial noise can cause drastic failures in quadrupedal robots by exploiting weaknesses in the decision-making processes of learning-based controllers. While the impact of these attacks has been explored, detection mechanisms that can robustly identify and mitigate these perturbations in real-time remain underdeveloped. This paper proposes a novel hybrid approach that combines a Kalman filter with a Physics-Informed Neural Network (PINN) to continuously monitor sensor inputs and controller outputs.(5) By integrating contextual information such as contact surfaces, speed, and trajectories, we aim to develop a robust knowledge base capable of detecting adversarial attacks across a variety of locomotion tasks.

Our approach leverages the complementary strengths of model-based and learning-based techniques to provide real-time, adaptive attack detection. The Kalman filter facilitates state estimation and smoothens out noise while ensuring adherence to expected physical behaviors. Simultaneously, the PINN incorporates the underlying physics of the robot's locomotion to enforce physically plausible predictions across different tasks, from walking to running. By capturing deviations in sensor data and controller output that violate physical laws or expected patterns, our system can effectively detect adversarial noise-based attacks. This integrated detection framework advances the current state-of-the-art by not only identifying perturbations but also providing a deeper understanding of how these attacks influence the robot's behavior under diverse operating conditions.

Our contributions are:

- **Hybrid Detection Framework:** We introduce a hybrid detection framework that combines a Kalman filter with a Physics-Informed Neural Network (PINN) to detect adversarial attacks on quadrupedal robots during locomotion tasks. The source code for this work is available on GitHub.

- **Contextual and Task-Specific Monitoring:** Our approach captures contextual information such as contact surfaces, speed, and trajectory data to build a comprehensive knowledge base, enhancing detection across a variety of locomotion tasks (e.g., walking, running).

- **Real-Time Detection of Adversarial Perturbations:** We demonstrate real-time detection capabilities by leveraging the Kalman filter for state estimation and the PINN for enforcing physically plausible behavior across diverse operating conditions.

- **Improved Robustness Against Attacks:** By integrating both model-based and data-driven techniques, we improve the robustness of quadrupedal robots against subtle adversarial perturbations that exploit the vulnerabilities of learning-based controllers.

- **Empirical Validation:** We validate our approach through extensive experiments across multiple tasks, showing how the proposed method effectively identifies adversarial attacks in terms of accuracy and robustness.

2 Background

Quadrupedal robots have gained significant attention due to their ability to navigate complex terrains and perform tasks requiring high mobility. These robots rely on advanced control algorithms for their locomotion, often employing machine learning techniques, particularly reinforcement learning (RL) (6), to improve their efficiency and adaptability in real-time environments.

Learning-based controllers, especially those utilizing deep neural networks (DNNs) and reinforcement learning, have shown great promise in enhancing the locomotion capabilities of quadrupedal robots (7).

However, these systems are vulnerable to adversarial attacks, where small, carefully crafted perturbations to the input data can significantly degrade the performance of the system Such vulnerabilities present serious risks in real-world applications, where robustness and reliability are crucial.

Adversarial attacks in robotic control systems can manifest in different forms, including attacks on sensor inputs, actuator commands, or the model itself. Several studies have proposed methods for detecting and mitigating these attacks. However, most of these methods are not well-suited for real-time deployment in robotic systems, as they tend to be computationally expensive or inaccurate under time constraints.

3 System Design

3.1 High level Architecture

Figure 1 demonstrates the high level architecture of the controllers for Quadrupedal robots, and where our proposed approach will sit. The system is structured into three key layers: Cyber Layer, Device Control Layer, and Physical Layer. Each layer interacts with the others to achieve efficient and safe operation of a robotic system.

Cyber Layer: This layer provides centralized control and high-level task management. It oversees the overall operations and communicates commands to the lower layers. This centralized system can be deployed on a cloud or local server, ensuring scalability and enhanced processing capability.

Device Control Layer: At the core of the control operations, this layer consists of the on-board computer. This computer acts as an intermediary, receiving commands from the Cyber Layer and processing sensor inputs from the Physical Layer to determine the system's current state. This is where our proposed hybrid detection approach will sit and in the background detect unexpected behavior. The key components of this device controller are:

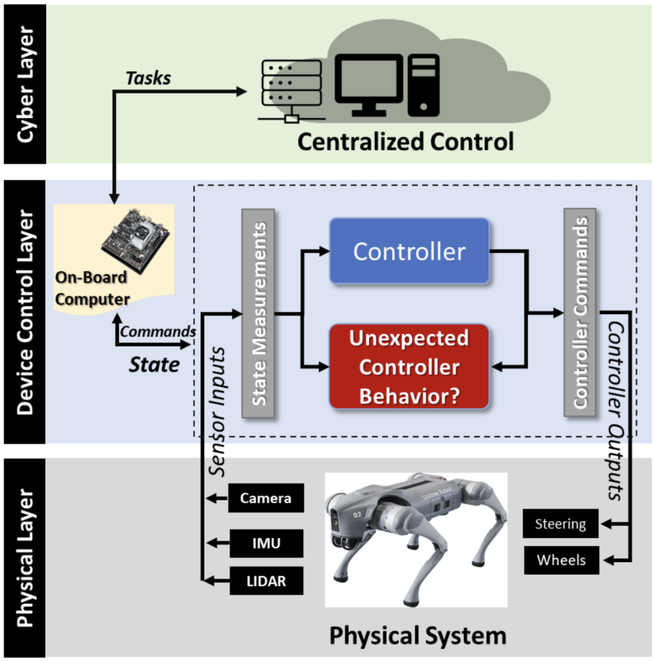

Figure 1: The high level system architecture of our system

- State Measurements: which measures and processes the various sensor inputs generating input "states" for downstream components
- Controller: Responsible for processing state measurements and issuing control commands to manage the physical system.
- Unexpected controller: In parallel to the controller, this component is responsible for processing both state measurements and controller commands to detect anomalies, such as unexpected controller behavior and ensure reliable operation. This is the core component where our hybrid approach will reside. See more details on how we detect such anomalies in Section 3.2

Physical Layer: This layer comprises the physical components of the robotic system, including sensors and actuators. The main elements are: 1) Sensors (Camera, IMU, LIDAR): Provide real-time data about the environment and system state. 2) Actuators (Steering, Wheels): Execute movement commands issued by the controller to achieve desired physical actions.

The Cyber Layer sends high-level tasks to the Device Control Layer, where the on-board computer processes inputs from the physical system and translates them into actionable commands, which are then then transmitted to the actuators in the Physical Layer, completing the loop. In tandem with this main loop, the unexpected controller monitors all inputs and commands to detect and handle unexpected controller behaviors, ensuring safe and robust system operation. This structure ensures modularity, fault tolerance, and adaptability for robotic systems.

3.2 Unexpected Controller

We propose a hybrid detection framework that leverages both model-based and data-driven approaches to achieve real-time detection of adversarial attacks on quadrupedal robots. The framework consists of two main components: a Kalman Filter (UKF) for state estimation and a Physics-Informed Neural Network (PINN) for enforcing physically plausible predictions.

3.2.1 unscented Kalman Filter (UKF) for State Estimation

The Kalman Filter (UKF) estimates the robot's state (e.g., position, velocity, and orientation) by smoothing noisy sensor data and providing an optimal estimate of the robot's true state. The Kalman Filter operates continuously during locomotion, enabling real-time monitoring across tasks and terrains.

3.2.2 Physics-Informed Neural Network (PINN)

The Physics-Informed Neural Network (PINN) incorporates physical laws and constraints into its learning process to ensure that the robot's predicted behaviors remain physically plausible. the PINN Embeds equations of motion and physical constraints (e.g., joint torque limits, friction coefficients) into the learning process. This ensures that predictions adhere to real-world physics, reducing false positives and improving detection accuracy.

3.2.3 Real-Time Detection Process

The hybrid framework continuously monitors the robot's behavior to detect adversarial attacks. In uses the input state measurements and generated commands as signals for this analysis. This real-time detection is composed of the following three steps as shown in Figure 2.

1. **State Prediction:** The Kalman Filter predicts the robot's state based on the current model, while the PINN predicts the next state based on physical laws. The integration of these two approaches leverages the strengths of both traditional sensor fusion and machine learning. As shown in Figure 3, the proposed PINN-UKF framework comprises three modules: the sensors module, the PINN module, and the UKF module.

   The sensors module integrates data from various sources, including GNSS, IMU, and a Virtual Sensor. These inputs serve as the foundation for accurate state estimation.

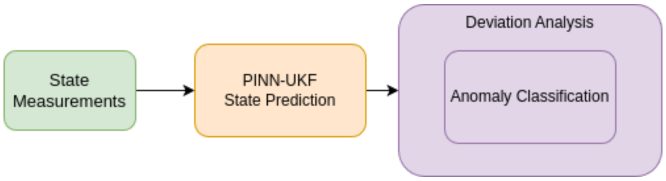

Figure 2: Real-time detection framework for monitoring robotic behavior and detecting adversarial attacks.

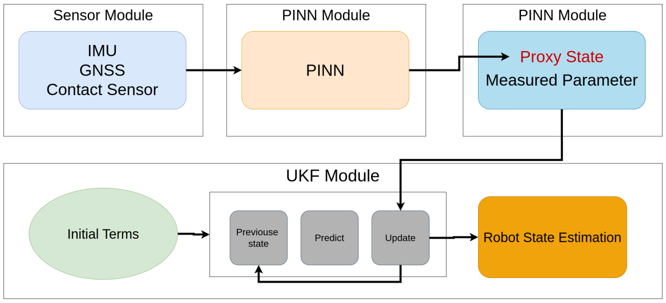

Figure 3: The structure of the proposed robot state prediction

   The PINN module is designed to handle noisy sensor data by learning the nonlinear relationships between sensor signals and the filtered robot states. Time-series sensor signals are processed as inputs to this module, which then generates proxy-states that act as refined approximations of the robot's state. These proxy-states serve as intermediate representations that are critical for robust state estimation.

   Finally, the UKF module processes the proxy-states generated by the PINN module. Using a state-space model, it refines these estimates to produce accurate predictions of the robot's position, velocity, and orientation.

   The combined use of PINN and UKF in the proposed framework ensures precise and robust state estimation by taking advantage of both physical modeling and data-driven learning, as depicted in Figure 3.

2. **Deviation Analysis:** Compares predicted states (from PINN-UKF) with actual sensor data. Significant discrepancies signal potential anomalies.

3. **Anomaly Classification:** Contextual data (terrain type, speed, trajectory) is used to classify anomalies as adversarial attacks or benign deviations. This is a critical component to avoid false positives, since some level of deviation is expected in a real scenario. Thus, we have developed an anomaly classification that can classify the expected deviations from unexpected anomalies.

4 Implementation

The success of such a controller is dependent on having a rich, diverse dataset of attacks to train the Kalman Filter and PINN effectively. Towards this, we have used a simulation environment based on multiple simulators and generated a rich dataset of both normal and adversarial conditions. We use a reinforcement learning approach as a Adversarial Perturbation Agent to generate realistic attacks and generated commands/inputs. The remainder of this sections describes both the dataset collection and the adversarial perturbation agent.

4.1 Dataset Collection

We developed a diverse dataset of quadrupedal robot trajectories using simulators like Isaac Gym(8)(9), PyBullet, and MuJoCo(10). The dataset includes normal conditions as well as controlled perturbations designed to simulate realistic disturbances.

4.1.1 Normal and Adversarial Conditions

Normal Conditions involve standard terrain and sensor accuracy, providing a baseline for the robot's stable behavior.

Adversarial Conditions simulate different disturbances, including noise injection to mimic sensor and actuator noise, external perturbations that represent impacts or environmental interferences, varied terrain textures with changes in ground textures and slopes, and intentional adversarial attacks targeting control signals.

4.2 Adversarial Perturbation Agent

We develop a reinforcement learning (RL) agent to generate a rich dataset of attacks. The agent creates adversarial attacks against the robot, introducing noise or confusing input signals and collects the generated commands by the robot as a baseline dataset, without the unexpected controller.

Our RL agent is tailored to create a range of adversarial scenarios by varying perturbations across different time frames—short-term and long-term—to effectively test the robot's resilience. This design challenges the agent to explore a range of destabilizing conditions, ensuring that it generates perturbations with diverse impacts on the robot's trajectory and stability. The reward function R_t at time t is:

$$
\begin{aligned}
R_t = & \, w_{\text{short}} \sum_{k=t}^{t+N_{\text{short}}} \left(w_{\text{pos\_short}} \|\Delta \mathbf{p}_k\| + w_{\text{orient\_short}} \|\Delta \theta_k\| \right) \\
& + w_{\text{long}} \sum_{k=t}^{t+N_{\text{long}}} \left(w_{\text{pos\_long}} \|\Delta \mathbf{p}_k\| + w_{\text{orient\_long}} \|\Delta \theta_k\| \right) \\
& + w_{\text{torque}} \sum_j \text{ReLU} \left(\frac{|\tau_j|}{\tau_{\text{lim},j}} - 1 \right) - w_{\text{reg}} \|\mathbf{a}_t\| + w_{\text{fail}} \mathbb{I}_{\text{fail}}.
\end{aligned} \tag{1}
$$

In this formulation:

w_{short} and w_{long}: Weights for short- and long-term planning horizons.

$\Delta \mathbf{p}_k$ and $\Delta \theta_k$: Position and orientation deviations at step k, scaled by corresponding weights $w_{\text{pos\_short}}, w_{\text{orient\_short}}, w_{\text{pos\_long}}, w_{\text{orient\_long}}$.

Torque term: Includes a ReLU penalty for exceeding joint torque limits, encouraging realistic perturbations.

Failure term $w_{\text{fail}} \mathbb{I}_{\text{fail}}$: Rewards the agent for achieving destabilization, reinforcing the failure objective.

Our Adversarial Perturbation Agent is inspired by the work of Dobrovolskiy et al.(11)

5 Experimental Evaluation

The main goal of our evaluation our project is to answer:

- what is the accuracy of the UKF estimation?
- how resilient is this approach to different tasks and terrains?
- how accurately can we identify true attacks.

We have currently performed a preliminary evaluation by focusing on the first two question, measuring and analyzing the accuracy of the UKF estimation. We are still working on further analysis of the attack detection and that will be part of future work.

5.1 Experimental Setup

To assess the performance of our hybrid detection framework, we tested it across various terrains and locomotion tasks. The tasks include walking in a straight line, random walking, and goal-reaching, under both normal and adversarial conditions. We used a comprehensive dataset collected through simulators.

We collect data under two scenarios:

- **normal**, represented as True [*metric*] showcasing the value collected by the simulator of a given metric of interest with no attacks.

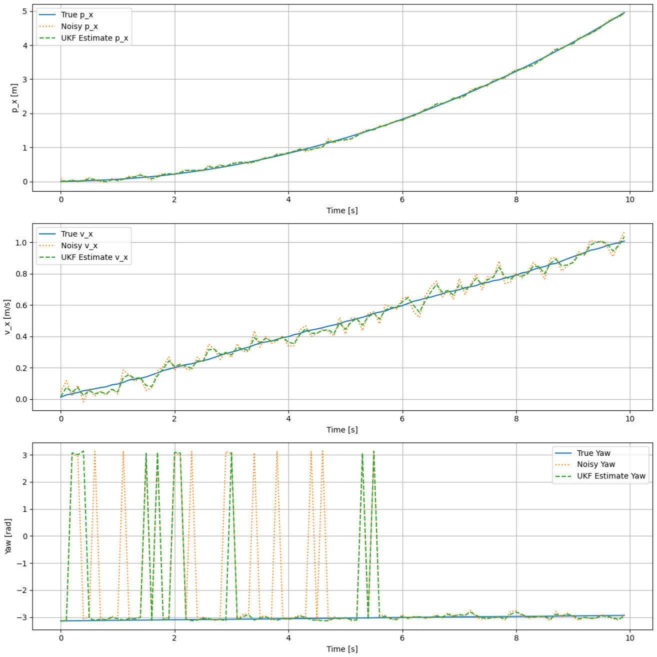

Figure 4: Prediction of the Unscented Kalman Filter (UKF) in real-time during robot locomotion, comparing True (no attack), Noisy (under attack measurement), UKF Estimation (under attack prediction) for various metrics.

- **under attack**, represented as Noisy [*metric*] showcasing the value collected by the simulator of a given metric of interest under a known attack.

For each of these scenarios, we collect various metrics from the robot and compare the results. For instance, we measure the following:

- *p_x* the position of the robot in the x-axis.
- *v_x* the speed of the robot in the x-axis.
- *Yaw* the rotation of the robot in the z-axis.

5.2 UKF Prediction Accuracy

We measure the effectiveness of the Unscented Kalman Filter (UKF) in state estimation by comparing the predicted vs. actual states of the robot during a typical locomotion task, both with and without attacks, shown in Figure 4. This figure compares the three scenarios True (no attack), Noisy (under attack measurement), UKF Estimation (under attack prediction) for various metrics of interest: position (p_x), speed (v_x), and Yaw.

First, we can see the introduction of noise by our RL based agent can effectively create sufficient noise on many metrics, creating the potential opportunity to identify such unexpected patterns and anomalies. Over time, we created a diverse series of such attacks.

Secondly, on certain metrics such as position and speed, the Kelmen filter is very effective, robust, and accurate at predicting the next values. This demonstrated the feasibility of relying on Kelmen filters to accurately predict the environment. While this does not generalize to all metrics, e.g., Yaw, having a few metrics that can uniquely identify attack is sufficient to build the overall unexpected controller. As a future step we are analyzing the exact accuracy of the overall unexpected controller.

6 Conclusion & Future Work

In this paper, we presented a novel hybrid detection framework for adversarial attack mitigation in learning-based locomotion controllers for quadrupedal robots. By combining the strengths of a Kalman Filter and a Physics-Informed Neural Network (PINN), our approach ensures robust and real-time detection of unexpected anomalies while adhering to physical plausibility constraints. Our preliminary simulation experiments demonstrated that our method significantly reaches high accuracy, robustness, and adaptability across diverse terrains and tasks in predicting the next move in noisy environments.

Our contributions underscore the importance of integrating model-based and data-driven techniques to address the vulnerabilities of learning-based controllers in dynamic and adversarial scenarios. Beyond detection, our framework provides insights into the influence of adversarial perturbations, paving the way for enhanced resilience in robotic systems.

Future work will focus on extending this approach to analyzing the accuracy of attack detection, testing it in real-world scenarios, and exploring adaptive mechanisms that allow robots to autonomously recover from adversarial influences.

References

[1] J. Li, Y. Liu, T. Chen, Z. Xiao, Z. Li, and J. Wang, "Adversarial attacks and defenses on cyber–physical systems: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5103–5115, 2020.

[2] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," *arXiv preprint arXiv:1712.03632v1*.

[3] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *International conference on machine learning*. PMLR, 2017, pp. 2817–2826.

[4] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell, "Adversarial policies: Attacking deep reinforcement learning," *arXiv preprint arXiv:1905.10615v3*.

[5] Y. Liu, Y. Bao, P. Cheng, D. Shen, G. Chen, and H. Xu, "Enhanced robot state estimation using

physics-informed neural networks and multimodal proprioceptive data," in *Sensors and Systems for Space Applications XVII*, vol. 13062. SPIE, 2024, pp. 144–160.

[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. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, "Learning-based legged locomotion; state of the art and future perspectives," *arXiv preprint arXiv:2406.01152*, 2024.

[8] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470v2*.

[9] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[10] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive Sampling: Real-time Behaviour Synthesis with MuJoCo," dec 2022. [Online]. Available: https://arxiv.org/abs/2212.00541

[11] F. Shi, C. Zhang, T. Miki, J. Lee, M. Hutter, and S. Coros, "Rethinking robustness assessment: Adversarial attacks on learning-based quadrupedal locomotion controllers," *arXiv preprint arXiv:2405.12424v2*.