System Overview

This project focuses on developing a reinforcement learning (RL) policy that enables a quadruped robot to navigate unpredictable and hazardous environments where human exploration is risky or impractical. The main goal of the RL is to enable the robot to autonomously adjust its movements based on environmental conditions, allowing it to traverse uneven, deformable, or obstructed terrain without reliance on pre-defined gaits.

State Representation

The robot's decision-making is based on a set of state variables that provide information about its position, motion, and interaction with the terrain:

- **Root orientation** – Measures the robot's roll and pitch to track its balance.
- **Root angular velocity** – Captures rotational speed across roll, pitch, and yaw to regulate excessive turning.
- **Root linear velocity** – Estimates translational motion to maintain smooth movement.
- **Joint angles and joint velocities** – Monitor limb positioning and movement speed.
- **Binary foot contacts** – Detects which feet are in contact with the ground for stability assessment.
- **Previous action** – Provides historical action data to improve policy consistency.
- **Linear velocity estimate** – Uses a Kalman filter combining acceleration and leg velocity to approximate motion, ensuring reliable state feedback in dynamic conditions.

Action Representation

The robot outputs joint targets as control actions to adjust its movement dynamically:

- **PD position targets for 12 joints** – Specifies desired positions for each joint.
- **Execution at 20Hz** – Updates control commands at a frequency of 20 times per second.
- **Per-leg action range** – Defines joint target limits based on an initial pose with an adjustable offset.
- **Position-based torque control** – Uses proportional and derivative gains to regulate motion.
- **Reset policy** – Implements an open-source reset mechanism for real-world operation when the robot loses stability.

Simulation and Deployment

The robot is tested across five terrains that simulate real-world hazardous conditions:

1. **Flat solid ground** for controlled stability assessment
2. **Memory foam** for soft, unstable terrain similar to snow or mud
3. **Mulch** to represent obstructive debris fields
4. **Lawn** for variable traction environments

5. **Hiking trails** with uneven surfaces, inclines, and natural obstacles.

Training occurs in both simulation and real-world settings, with data collected at **1000 time-steps per minute** using randomly sampled actions. The system continuously trains until stability thresholds are exceeded, at which point a reset policy is used.

Your Task

You will contribute to designing the system with a focus on **[OBJECTIVE]**. To do this, you will complete the following steps:

Steps to Complete:

1. Read through the **Task Overview** and **System Assumptions** (available here: [Assumptions]).
2. Define System Requirements for [OBJECTIVE]:
   a. Identify key system requirements necessary to achieve **[OBJECTIVE]** within the system.
   b. Use the following [EARS] format for your response:
      - When <optional trigger>, the <system name> shall <system response>
   c. For each requirement, detail a **potential design solution** in bullet points.

   Example:

   - **Requirement:** *When a sudden obstacle is detected, the autonomous vehicle shall apply emergency braking to avoid a collision.*
   - **Potential Design Solutions:**
     - Implement LiDAR and camera-based object detection to identify obstacles in real time.
     - Develop an emergency braking algorithm that calculates safe stopping distances.
     - Integrate a redundant braking system to ensure functionality in case of sensor failure.
     - Conduct extensive simulations and real-world testing to validate emergency braking effectiveness.
3. Answer RL-Related Questions:
   a. After completing the requirements, you will receive an Excel sheet containing a set of questions assessing the RL & system design.
   b. Provide **detailed responses**, following the provided example answer as a guide for the expected level of specificity:
      - Collision avoidance accuracy measures the percentage of successfully detected and avoided obstacles, with an acceptable range of ≥98% based on autonomous flight performance benchmarks. Evasion precision is assessed using the Mean Absolute Error (MAE) of deviation from the

optimal avoidance path, with an acceptable deviation of ≤0.5 meters to ensure precise maneuvering. These thresholds are determined through industry standards and empirical performance evaluations.

   c. Some high-level questions (marked with an asterisk *) are broken down into sub-questions which may be fully addressed by responding to all sub-questions. If so, you may skip the high-level question as long as your responses to the sub-questions fully address it.
   d. Some of your responses may overlap—this is okay! If responses overlap, you may **reference previous answers** (include the relevant ASK ID #) but you should still explain how the design choice applies to the current question.
   e. If a question does not apply, respond with **N/A** and briefly explain why.
4. Review and Provide Feedback on Requirements:
   a. Based on your responses, a refined set of system requirements will be developed.
   b. You will review these new requirements and provide feedback in a **brief interview**.
   c. During this session, you may be asked to **elaborate on some of your responses**.

Additional Notes

- You may use any available tools to assist with your task.
- However, you must be able to **explain and justify** your answers in a short meeting.
- Do not submit any responses you **cannot clearly explain**.
- You will **not** be implementing the system, but your responses should be **realistic and implementable** given sufficient time.