Contents

1. Summary

1.1 Background

The Spot Robot Reinforcement Learning with Augmented Random Search (ARS) project focuses on training a Spot robot using reinforcement learning techniques. The robot's movements are simulated and optimized through the ARS algorithm.

1.2 Objectives

- Train a reinforcement learning agent to control a Spot robot using ARS.
- Investigate the impact of various training configurations on agent performance.
- Evaluate the adaptability and navigation capabilities of the trained agent.

1.3 Scope

The project includes the 3D printing of the Spot robot model and acquiring necessary components to replicate a physical manifestation of the robot for future physical testing.

2. Implementation Details

2.1 Physical Spot Robot

Before diving into the reinforcement learning aspects, a physical Spot robot was created by 3D printing the model and procuring essential components.

2.1.1 3D Printing

The Spot robot model was 3D printed to provide a tangible representation of the simulated robot. This physical model allows for a deeper understanding of the robot's structure and facilitates future physical testing.

2.1.2 Components Acquisition

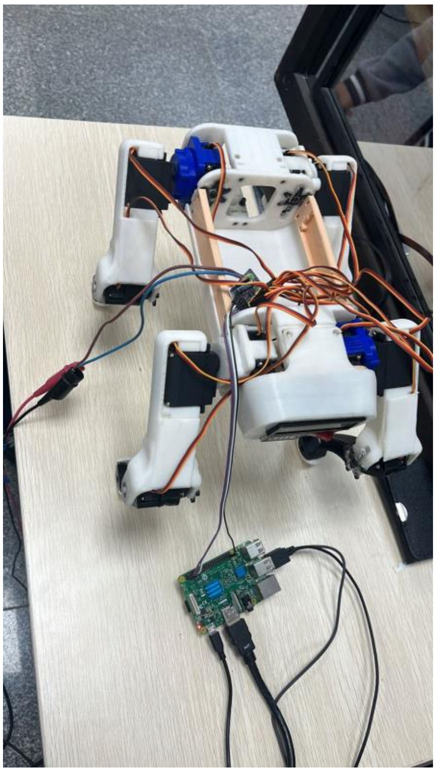
Various components, including motors and sensors, were acquired to build the physical Spot robot. These components form the hardware foundation for the robot's movements.

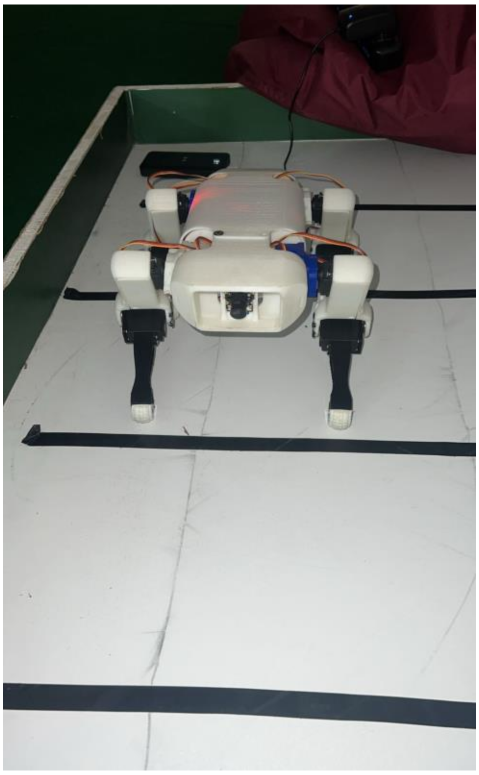
2.2 Progression Pictures and Video Presentation

The first Image is a picture of our robot being assembled.

The second image is a picture of our completed robot.

Lastly, Watch the Spot robot learning to adapt in its environment in this video presentation





3. Project Overview

3.1 Physical Spot Robot

Before diving into the reinforcement learning aspects, a physical Spot robot was created by 3D printing the model and procuring essential components.

3.1.1 3D Printing

The Spot robot model was replicated using 3D printing technology. This physical model serves as the tangible representation for the reinforcement learning experiments.

3.1.2 Components Acquisition

Necessary components, such as motors, sensors, and other electronic parts, were acquired to enable the Spot robot's movements.

3.2 Training Script (spot_ars.py)

The training script facilitates the training of the Spot robot reinforcement learning agent using the ARS algorithm.

3.2.1 Training Process

Utilizes ARS algorithm for efficient training.

Models are saved every 9th episode in the spot_bullet/models/ directory.

3.2.2 Multiprocessing

Enhances training efficiency through multiprocessing.

Utilizes concurrent execution of multiple parallel workers.

3.2.3 Command-line Arguments

-hf or --HeightField: Use HeightField for training.

-nc or --NoContactSensing: Disable Contact Sensing during training.

-dr or --DontRandomize: Do NOT Randomize State and Environment.

-s or --Seed: Seed for randomization during training.

3.3 Evaluation Script (spot_ars_eval.py)

3.2.1 Evaluation Process

- Involves loading a pre-trained agent for evaluation.
- Visualization options include plotting policy output and true actions if specified.

3.2.2 Data Saving and Visualization

- Allows for saving policy output data in the results folder.
- Visualization features provide insights into the policy output and the robot's true actions during evaluation.

3.2.3 Command-line Arguments

-a or --AgentNum: Specify the Agent Number to load for evaluation.

-pp or --PlotPolicy: Plot the Policy Output after each Episode.

-ta or --TrueAction: Plot the Action as seen by the Robot.

-save or --SaveData: Save the Policy Output to a .npy file in the results folder.

4. Reinforcement Learning

training Steps:

1. Initialize Environment: Set up the simulation environment for the Spot robot.

2. 3D Print Model & Acquire Components: Physically replicate the Spot robot using 3D printing and acquire necessary components for the hardware.

3. Initialize Agent: Create a reinforcement learning agent, initializing its policy and normalizer.

4. Training Loop: Repeat the following steps until a specified number of episodes or a convergence criterion is met:

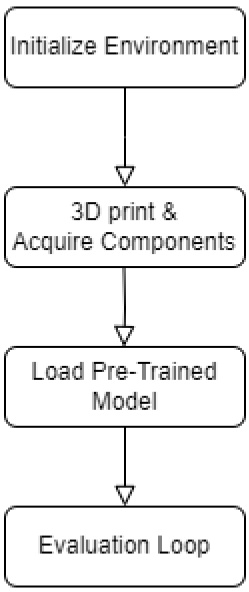
   4.1. Generate Perturbations: Generate perturbations to the policy parameters.

   4.2. Evaluate Perturbations: Evaluate the perturbed policies in the environment.

   4.3. Compute Rewards: Calculate the rewards obtained by the perturbed policies.

   4.4. Update Policy: Update the policy parameters based on the rewards using the ARS algorithm.

   4.5. Save Model: Periodically save the trained model.



*Evaluation **Steps:***

1. Initialize Environment: Set up the simulation environment for the Spot robot.

2. 3D Print Model & Acquire Components: Physically replicate the Spot robot using 3D printing and acquire necessary components for the hardware.

3. Load Pre-Trained Model: Load the pre-trained reinforcement learning model for evaluation.

4. Evaluation Loop: Repeat the following steps for each episode:

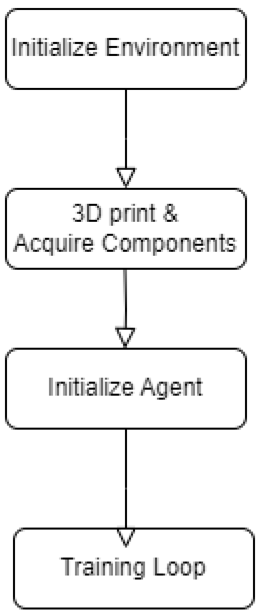
   4.1. Execute Policy: Execute the policy to generate actions in the environment.

   4.2. Observe State and Reward: Observe the state and reward obtained from the environment.

   4.3. Record Actions and States: Record the actions executed and states observed during the episode.

   4.4. Visualize Results (Optional): Optionally, visualize the robot's actions and states.

   4.5. Save Policy Output (Optional): Optionally, save the policy output data.



5. References

1. Inspiration and CAD files from the Spot Micro Ai Community
2. Modified CAD files and PCB design Adham Elarabawy
3. OpenAI Gym Interface: Miniature Environment
4. URDF files: Rex Gym