GoatBot: A High-Speed, Sustainable, Incline-Balancing Mountain Goat Robot

1 Overview

GoatBot is an open-source, 3D-printed, four-legged robotic platform inspired by Kawasaki's CORLEO robotic horse but designed to surpass it in speed, sustainability, and autonomy. Built with recycled materials, a Raspberry Pi-based touchscreen interface, and a high-capacity lithium-ion battery, GoatBot achieves speeds of 65–70 mph, excels on rugged inclines, and features an autonomous "return home" system for user safety. It uses biomimetic hooves with advanced traction, a lightweight rPETG chassis, and AI-driven terrain prediction to ensure stability and durability.

1.1 Key Features

- **High Speed**: Reaches 65–70 mph using brushless DC (BLDC) motors and aerodynamic design.
- **Superior Traction**: 3D-printed, dual-layer TPU lattice hooves inspired by mountain goats, optimized for rugged terrain.
- **Sustainability**: Built with recycled PETG (rPETG) and TPU, minimizing metal use and environmental impact.
- **Advanced Control**: Reinforcement learning (RL) balance controller and AI terrain prediction, running on a Raspberry Pi 4 with a touchscreen UI.
- **Autonomous Navigation**: GPS/RTK-based "return home" system for safe operation if the user is impaired.
- **Durability**: Honeycomb chassis, TPU suspension, and cooling ducts ensure longevity at high speeds.
- **Battery Efficiency**: 6S 8000mAh lithium-ion battery with regenerative braking, inspired by electric dirt bikes like the Segway X260.
- **Cost-Effective**: Uses refurbished components and recycled materials (~$500 total cost).

1.2 Comparison to Kawasaki's CORLEO

- **Speed**: 65–70 mph vs. CORLEO's estimated 30–40 mph.
- **Sustainability**: Recycled plastics and electric power vs. CORLEO's likely metal-heavy, hydrogen-powered design.
- **Control**: RL and AI-driven systems vs. CORLEO's simpler reactive controls.

- **Autonomy**: GPS/RTK return-home feature vs. CORLEO's lack of autonomous navigation.
- **Traction**: Multi-material lattice hooves vs. CORLEO's rubber hooves.
- **Cost**: ~$500 vs. CORLEO's estimated $10,000+.

2 Repository Structure

```
GoatBot/├──
 README.md├──
 LICENSE├──
 docs/│├──
    design_specification.md│├──
    assembly_guide.md│├──
    control_system_guide.md│├──
    battery_management.md│├──
    patent_workarounds.md│├──
    material_selection.md│├──
    testing_protocols.md│├──
    navigation_guide.md│├──
    thermal_management.md├──
 cad/│├──
    body/││├──
       main_chassis.stl││├──
       leg_mounts.stl││├──
       cooling_duct.stl│├──
    legs/││├──
       upper_leg.stl││├──
       lower_leg.stl││├──
       hoof.stl││├──
       suspension.stl│├──
    head/││├──
       pi_mount.stl││├──
       screen_holder.stl│├──
    fea/││├──
       leg_stress_analysis.fea││├──
       chassis_stress_analysis.fea││├──
       suspension_stress_analysis.fea├──
 src/│├──
    control/││├──
       main_control.py││├──
       balance_controller.py││├──
       terrain_adaptation.py││├──
       motor_driver.py││├──
       rl_model.py││├──
       terrain_prediction.py│├──
    interface/││├──
       touchscreen_ui.py││├──
       sensor_interface.py│├──
```

```
│   navigation/││├──
│      path_planning.py││├──
│      sensor_fusion.py││├──
│      gps_navigation.py├──
│ firmware/│├──
│   motor_firmware.ino│├──
│   bms_firmware.ino│├──
│   bldc_firmware.ino├──
│ bom/│├──
│   bill_of_materials.csv│├──
│   sourcing_guide.md├──
│ tests/│├──
│   unit_tests/││├──
│      test_balance.py││├──
│      test_terrain.py││├──
│      test_navigation.py│├──
│   integration_tests/││├──
│      test_full_system.py├──
│ scripts/│├──
│   setup_environment.sh│├──
│   flash_firmware.sh├──
│ configs/│├──
│   robot_config.yaml│├──
│   sensor_config.yaml│├──
│   navigation_config.yaml└──
│ media/├──
│    images/│├──
│       robot_render.jpg│├──
│       hoof_design.jpg│├──
│       suspension_design.jpg├──
│    videos/│├──
│       demo_walk.mp4│├──
│       incline_test.mp4│├──
        high_speed_test.mp4
```

2.1 File Descriptions

- **docs/**: Design specifications, assembly guides, control system details, battery management, patent workarounds, material selection, testing protocols, navigation guide, and thermal management.
- **cad/**: STL files for 3D-printed parts (chassis, legs, hooves, suspension, cooling ducts) and FEA files for stress analysis.
- **src/**: Python scripts for control (RL, terrain prediction), touchscreen UI, sensor fusion, and GPS navigation.
- **firmware/**: Arduino code for servo motors, BLDC motors, and battery management system (BMS).

- **bom/**: Bill of materials (BOM) and sourcing guide for components.
- **tests/**: Unit and integration tests for balance, terrain adaptation, and navigation.
- **scripts/**: Setup and firmware flashing scripts.
- **configs/**: Configuration files for robot parameters, sensors, and navigation.
- **media/**: Images and videos of renders, designs, and test runs.

3 Getting Started

3.1 Prerequisites

- **Hardware**:
  - Raspberry Pi 4 (4GB or 8GB)
  - 6S 8000mAh lithium-ion battery with BMS
  - Dynamixel AX-12A servos (front legs) and 1000W BLDC motors (rear legs)
  - RPLIDAR A1, stereo cameras, Ublox NEO-6M GPS module
  - 7-inch touchscreen display
  - 3D printer with rPETG and TPU filament
- **Software**:
  - Raspberry Pi OS (64-bit)
  - Python 3.8+, ROS2, RTKLIB
  - Arduino IDE for firmware
  - Cura slicer for 3D printing

3.2 Setup Instructions

1. **Clone the Repository**:
   ```
   git clone https://github.com/GoatBot/GoatBot.git
   cd GoatBot
   ```
2. **3D Print Parts**:
   - Use `cad/` STL files with rPETG (chassis, legs) and TPU (hooves, suspension).
   - Recommended settings: 0.2mm layer height, 20% infill, 3 perimeters.
3. **Assemble Hardware**:
   - Follow `docs/assembly_guide.md` for step-by-step instructions.
   - Mount Raspberry Pi, touchscreen, sensors, and motors.
4. **Install Software**:

   - Run `scripts/setup_environment.sh` to install dependencies on the Raspberry Pi.
   - Flash firmware to microcontrollers using `scripts/flash_firmware.sh`.

5. **Configure**:
   - Edit `configs/robot_config.yaml`, `sensor_config.yaml`, and `navigation_config.yaml` for your setup.

6. **Test**:
   - Run unit tests: `python -m unittest tests/unit_tests/*.py`.
   - Perform integration tests: `python tests/integration_tests/test_full_system.py`.
   - Test on inclines and at high speeds (see `docs/testing_protocols.md`).

4 Usage

- **Control**: Use the touchscreen UI (`src/interface/touchscreen_ui.py`) to monitor terrain, adjust gait, or initiate return-home mode.
- **Navigation**: Activate return-home via UI or app signal; configure home coordinates in `configs/navigation_config.yaml`.
- **Maintenance**: Replace modular legs (`cad/legs/`) and check cooling ducts (`cad/body/cooling_duct.s` regularly.

5 Patent Workarounds

To avoid infringing robotics and 3D printing patents:

- **Legged Robotics**: Uses open-source control algorithms (e.g., Solo 8) and custom 3D-printed joints to avoid patents like US10414039B2.
- **3D Printing**: Relies on expired FDM patents (e.g., US5121329A) and open-source slicers.
- **Hooves**: Unique TPU lattice design differentiates from CORLEO's rubber hooves (e.g., JP2021154920A).
- **Navigation**: Open-source RTKLIB for GPS/RTK avoids proprietary systems.

See `docs/patent_workarounds.md` for details.

6 Contributing

We welcome contributions! Please:

1. Fork the repository.
2. Create a feature branch (`git checkout -b feature/YourFeature`).
3. Commit changes (`git commit -m ”Add YourFeature”`).

4. Push to the branch (`git push origin feature/YourFeature`).
5. Open a pull request.

Follow the guidelines in `CONTRIBUTING.md` (to be created).

7 License

GoatBot is licensed under the BSD 3-Clause License. See `LICENSE` for details.

8 Acknowledgments

- Inspired by Kawasaki's CORLEO and open-source projects like Solo 8 and OpenCat.
- Thanks to the open-source community for ROS2, RTKLIB, and 3D printing resources.

9 Contact

For issues or questions, open an issue on GitHub or contact the maintainers at [hypothetical email]. Happy building!