**Project Documentation: MATLAB-Based Agricultural Rover Simulation**

**1. Introduction**

This project focuses on the development and simulation of an intelligent agricultural robotic rover within a high-fidelity 3D farmland environment using MATLAB. The primary goal was to investigate and implement autonomous navigation and terrain-leveling strategies under realistic field conditions. The simulation enables detailed analysis of rover performance over uneven terrain, leveraging algorithms for path planning, terrain modification, and multi-agent coordination. It acts as a testbed for validating various strategies used in precision agriculture without the need for physical prototyping.

**2. Objectives**

* Simulate natural farmland terrain with realistic noise and elevation profiles
* Model an agricultural rover with physically plausible movement and orientation
* Implement various path planning algorithms (A\*, spiral, rectangular spiral, multi-pass)
* Simulate terrain interaction through progressive height modification to replicate plowing behavior
* Create a modular simulation framework supporting multiple rover instances
* Provide real-time visualization of rover behavior, terrain status, and path efficiency metrics

**3. Simulation Environment Setup**

**Terrain Generation**

* A 5x5 meter terrain grid was initialized using meshgrid and populated with a base sinusoidal wave pattern.
* Multiple octaves of Perlin-like random noise were superimposed to create non-uniform, organic bumpiness.
* Gaussian smoothing via imgaussfilt ensured soft transitions and removed unrealistic spikes.
* Bumps were constrained to 0.5–3 cm in height and randomly distributed with 10–30 cm width.

**Rover Design**

* The rover was designed as a cuboid mesh (30x30x15 cm) using patch() and rendered using custom vertices and faces.
* Local coordinate frames were defined to enable computation of pitch, roll, and yaw during movement.
* The rover’s orientation was adjusted in 3D space using transformation matrices from makehgtform, applying dynamic rotation according to terrain gradient.

**4. Path Planning Algorithms**

**A\* Grid-Based Planning**

* The field was divided into grid cells based on plow diameter (0.2m).
* Standard A\* algorithm was customized with a heuristic that combined:
  + Distance to goal
  + Height difference from mean terrain level
  + Visit frequency penalties to encourage coverage
* Coordinate transformation corrected mismatches between planning and physical space.

**Spiral and Rectangular Spiral Planning**

* Parametric spirals were generated with linspace and scaled to fit field boundaries.
* Rectangular spiral paths were created with concentric inward-moving rectangles, using adjustable step sizes.
* Both strategies provided continuous, sweeping coverage, reducing overlaps and dead zones.

**Multi-Pass A\* Planning**

* Implemented a height-aware A\* pathfinder that calculated required passes per grid cell based on deviation from a target level.
* Cost function was dynamically updated to incorporate terrain variance, revisit penalties, and slope-based movement cost.
* Visit counters tracked the number of times each grid cell was covered to simulate iterative leveling.

**5. Terrain Interaction and Modification**

**Real-Time Plowing Simulation**

* A circular region of effect was defined around the rover using a plow mask.
* Each pass through a cell reduced height via an exponential decay model.
* A maximum reduction threshold per pass (~0.05m) was enforced to simulate realistic soil displacement.

**Plow Count Tracking and Visualization**

* A plow\_count matrix tracked cumulative passes per cell.
* Generated heatmaps using imagesc() to indicate terrain coverage.
* Combined with height maps, this gave insight into leveling uniformity and effectiveness.

**6. Multi-Robot Architecture**

**Modular Rover Class**

* Built a reusable MATLAB class encapsulating all rover functionality:
  + Position tracking
  + Path following with spline interpolation
  + Terrain data sampling
  + Visualization methods (color-coded rendering, orientation update)
* Each instance operated independently, allowing concurrent simulation of multiple units.

**Multi-Rover Coordination**

* A path segmentation algorithm divided a global path among available rovers.
* Rovers were instantiated in an array and looped through simultaneously.
* Data collection and plow effects were aggregated across all rovers to create a single terrain output.

**7. Visualization Features**

* Full 3D simulation with surf() for terrain, patch() for rover, and dynamic camera controls.
* Rover tilt and direction visualized in real time using transformation matrices.
* Dual-panel displays for terrain elevation and visit heatmaps.
* Live colorbars and legends dynamically updated based on simulation state.
* RViz-style marker overlays indicated key locations (start, goal, max plow count).

**8. Challenges Encountered**

* **Grid Coordinate Conversion Errors:**
  + Initial mapping from discrete A\* grid to continuous terrain space caused path misalignments.
  + Solved with cell\_size derivation and centralizing offset using -0.5 multipliers.
* **Path Redundancy and Missed Cells:**
  + Early paths had excessive overlaps and blind zones.
  + Resolved through adaptive cost functions and spiral/rectangular sweeping algorithms.
* **Rendering Performance Bottlenecks:**
  + High refresh rates and looped plotting slowed simulation.
  + Optimized with reduced plot update frequency and vectorized logic.

**9. Key Outcomes**

* Successfully simulated a dynamic plowing operation over realistic terrain with multi-pass optimization.
* Implemented and compared several intelligent path planning techniques under realistic rover constraints.
* Built a scalable and modular rover simulation system adaptable to other agricultural or industrial domains.
* Generated high-quality visualizations that can be directly used for presentation, analysis, or integration into a portfolio.

**10. Future Extensions**

* **Sensor Integration:**
  + Simulate GNSS and IMU modules with configurable noise models
  + Fuse sensor data using an Extended Kalman Filter (EKF)
* **ROS2 Transition:**
  + Port rover class and simulation framework to ROS2 with nav\_msgs, sensor\_msgs, and Gazebo integration
* **Realistic Agricultural Tasks:**
  + Add synthetic crop rows, obstacles, and planting modules
  + Simulate seed deposition or weed detection via virtual sensors
* **Optimization & AI:**
  + Add energy-aware path planners
  + Use Reinforcement Learning (RL) to tune leveling efficiency

**11. Conclusion**

The MATLAB-based agricultural rover simulation served as an end-to-end framework for exploring autonomous navigation, terrain modeling, and precision leveling. By combining advanced path planning, dynamic terrain manipulation, and modular scalability, it provided a cost-effective virtual testbed to validate intelligent farming algorithms. The project not only laid the groundwork for real-world autonomous farming systems but also demonstrated that high-fidelity simulation using MATLAB can effectively replicate and analyze complex agri-robotic workflows.

**Author:** Adharsh Prasad Natesan  
**Project Period:** Fall 2024 – Spring 2025  
**Environment:** MATLAB R2023a  
**Repository:** [Add GitHub URL here if applicable]