
A SYSTEMATIC REVIEW OF UW ROBOTICS ROVER DESIGN

Yuchen Lin
UWaterloo robotics team
University of Waterloo
Waterloo, ON
`{y29lin}@uwaterloo.ca`

1 Introduction

This documentation provides an in-depth technical review of the UW Robotics team's development process for our new rover platform, Sparky. The objectives of this documentation are as follows:

1. Provide in-depth analysis of each competition mission
2. Evaluate possible technologies and solutions proposed to address each mission
3. Document the rover design, system characterization, and design variants
4. Provide a reference manual for future UW Robotics members to reproduce the platform with a comprehensive understanding of system capabilities

This document is organized into the following sections:

- Problem Definition
 - Mission Requirements
 - Mission Analysis
 - Solution Proposal
- Environment Setup
 - Development and DevOps Setup
 - System Setup
 - Testing and Validation
- System Design
 - Baseline Performance
 - Design Notes
 - Implementation Details
- Maintenance Log

All systems are deployed with appropriate isolation. We emphasize utilizing technologies such as `python-venv` and `conda`. Additionally, following the practices established in existing Python projects, project folders should adhere to the template structure shown in Figure 1.

In summary, this documentation serves as a knowledge base and reference for future team members interested in contributing to the project.

2 Problem Definition

Season 2026, UW Robotics team are going to participate in University Rover Challenge(URC) and Canadian International Rover Challenge(CIRC)[1, 2].

```
project/
└── .git/workflows/
└── src/
    ├── nodes/      (ROS nodes)
    └── lib/        (algorithms)
    ├── inc/        (C++ headers)
    ├── mod/        (Python modules)
    ├── common/     (baseline implementations & metrics)
    ├── config/     (YAML/JSON configuration)
    ├── tests/      (unit & integration tests)
    ├── thirdparty/ (ported dependencies)
    ├── docs/
    └── launch/     (ROS launch files)
└── .gitignore
└── CMakefile, CMakeLists.txt, package.xml, setup.py
└── README.md
```

Figure 1: Project structure for UWaterloo robotics projects. This structure derived from standard ROS conventions with support for mixed Python and C++ development for any projects.

2.1 Mission Requirements

2.1.1 Setup, Transportation and Cleanup Requirement

Competition rules required the following:

•

2.2 Mission Analysis

2.2.1 Setup and Cleanup Requirement

Our team internally want this to be as minimal as possible, we went up the field, plug in and then good to go.

2.3 Solution Proposal

3 Maintenance Log

Date	Version	Changes
2025-11-26	1.0	Initial draft created

Table 1: Document revision history

References

- [1] CSTAG. Canadian intelligent robotics challenge, 2025. Accessed on November 26, 2025.
- [2] The Mars Society. University rover challenge, 2025. Accessed: 2025-11-26.