Design Draft for Rover Master

Haoyuan Xu
University of Florida
haoyuan.xu@ufl.edu

Jingxuan Xiao
Univerisyt of Florida
j.xiao@ufl.edu

***Abstract*—This design draft includes an introduction to our robot design, our expected progress at different stage, and the final deliverable artifact.**

Keywords—open-source, autonomous, battery management system, modular design, machine learning

I. Introduction

A. *Purpose*

Robots for educational purposes are extremely expensive with limited capability. A robot base can cost thousands or even tens of thousands of dollars to purchase when researchers need to perform computer vision, agricultural, human-robot interaction, or navigation projects. The purpose of our robot is to provide students and researchers with an open-source and well-rounded solution that is budget friendly. We want to create standards for different modules of our robot base so people can provide their innovations in this open-source community.

B. *Domain*

The Rover Master robot belongs to multiple overlapping domains, including Mobile Robotics, Educational Technology, and Research and Development, due to its design as an affordable, open-source, and customizable robotics platform suitable for students, educators, and researchers. It also falls under Computer Vision and Machine Learning because of its features like RGBD camera integration, depth estimation, and object detection, which are essential for advanced perception tasks. As an Autonomous System, Rover Master uses sensors such as IMU, optical flow sensors, and Lidar to navigate and make decisions independently. Its Modular Robotics nature allows users to customize and expand its functionality, which is ideal for R&D and prototyping. Additionally, Rover Master can be applied in Agricultural and Industrial contexts for automation, monitoring, and other practical applications, making it a versatile tool for various fields.

C. *Prior Art*

1) *MyAGV*

- Price: $1756
- Onboard Computer: Nvidia Jetson Nano 4GB
- Core Control: ESP32
- Sensors: Unknown Lidar
- Accessories: RGBD camera, touchscreen display, 6400mAh battery
- Drive train: Mecanum Wheels + Brushless DC motors
- Charging: Barrel Jack 12.6V 2A
- Customization: Mounting holes for robot arm

Figure 1: myAGV from Elephant Robotics

2) *Turtle Bot 4*

- Price: $2191
- Onboard Computer: Raspberry Pi
- Core Controller: Unknown
- Accessories: Charging Dock, RGBD camera
- Sensors: bumper, IR cliff sensor, IR obstacle sensor, optical flow sensor, IMU, RP Lidar
- Drive train: 3 wheel differential drive
- Charging: Charging Dock
- Customization: No official support

Figure 2: Turtle Bot 4

3) *Rover Master*

- Onboard Computer: Raspberry Pi 5 / Jetson Nano / Rock Pi 5
- Core controller: stm32
- Accessories: RGBD camera
- Sensors: optical flow sensor, IMU, RP Lidar.

- Drive train: Mecanum wheels / customized wheel + brushless DC motor + duo-suspension per wheel.
- Charging: PD charging at 20V 3-5A
- Customization: customizable size of aluminum plate, wheels, and mounting holes

Figure 3: Rover Master

Advantages:

- The 6 port usb controller on the main control board will allow more sensors to be connected.
- The wheel and suspension design allows it to handle more complex terrains.
- The charging method is more flexible and convenient. The battery is fixed on the robot, so no extra charging tools are needed.
- Modular design allows more customization and innovation.

Drawbacks:

- The ideal of having everything on one board makes integration a complicated task.
- Most materials and components are not commercially available, so users need to obtain them from various sources like 3D-printing, aluminum plate milling, and PCB manufacturing.
- The suspension design means the calibration of suspension will directly affect the posture and path of the robot when moving.

D. *Impact & Risk Assessment*

1) Cultural Impact

By providing affordable and open-source robot solutions, the robot can significantly broaden access to robotics research for people from all backgrounds. Students and researchers from underfunded schools or developing regions will have the opportunity to work with advanced robotics technology.

2) Global Collaboration

The open-source nature of the robot encourages collaboration across countries and continents. Researchers and developers can contribute to and learn from the community regardless of their geographic location.

3) Sustainable Robotics Design

The robot can be easily repaired and upgraded due to its modular design. By standardizing components and supporting customization, users can replace parts or upgrade systems rather than discarding the entire robot.

4) Ethical context

Open-source projects present risk in terms of security. There is a possibility that users could modify the robot for unethical purposes. Clear guidelines should be established to minimize misuse of the platform for unethical activities or crimes.

5) Economic Displacement

As robots become more affordable and widely available, there may be concerns about the displacement of human workers in agriculture, manufacturing, and other fields. It will be essential to address how the robot can complement, rather than replace, human labor in certain applications.

II. State of Work

A. *Hardware – Haoyuan Xu*

1) Features
   - Main control board can stack up with either raspberry pi or jetson nano for compact size.
   - Integrated battery system with built-in active battery balancing circuit that supports PD charging.
   - Onboard IMU for localization and posture sensing.
   - Stepper motor controller powered by esp32 Arduino nano.
   - Voltage or current sensing for most power outputs.
   - 1 to 6 USB controller for sensor expandability.
   - Dshot signal to electric speed controller for brushless DC motor control.
2) Tasks
   a) main control board with Pi 5 integration
      - This includes the 1 to 6 USB controller and USB to serial circuit design for sensor compatibility. Optical flow sensors and IR sensors can be implemented if desired.
      - The GPIO pins can be connected through Board to Board connector or spring-loaded pins.
   b) Battery Management System
      - The robot features a type C charging with PD negotiation capability.
      - The active balance charging is achieved by quickly charging and discharging between the battery and capacitors.
   c) Stepper controller board
      - Stepper drivers are controlled by esp32 Arduino nano.
      - TMC2209 drivers will be used to convert dir and step signals to 4 wire stepper signals.
   d) Firmware development

- Stepper controller: Firmware development for stepper controller.
- Main control board: Firmware development for imu integration and voltage/current sensing, and Dshot signal for brushless DC motor control.

3) Milestones

a) Semester One

Week 1-2: 10/7 - 10/20

- PCB design for main control board.
- The PD controller circuit will be designed but not integrated at this stage.
- Balance charging will be designed on a separate board for safety consideration.

Week 3-4: 10/21 - 11/3

- Wait for PCB production.
- Keep refining the PCB design, especially battery balancing.
- Start firmware development for stm32 on the main control board.

Week 5-6: 11/4 - 11/17

- Test the PCB and revise the design
- Continue the firmware development for the main control board.
- Start to design the stepper motor control board.

Week 7-8: 11/18 - 12/1

- Continue working on the stepper motor control board.
- Continue the firmware development for the main control board if necessary.
- Test the battery management system.

b) Semester Two

Week 7-8: 1/13 - 1/25

- Integrate the battery management system on the main board.
- Wait for the stepper driver control board PCB.
- Start firmware development for stepper control board.

Week 9-10: 1/27 -2/8

- Test the stepper control board on the robot
- Do research on the electric speed controller.

Week 11-12: 2/10 - 2/22

- Integrate the stepper control board with the main system
- Continue research into electric speed controllers
- Begin initial testing and calibration of the peripheral sensors

Week 13-14: 3/24 - ⅘

- Design the electric speed controller circuit
- Begin testing and calibration of the drivetrain with Mecanum wheels
- Fine-tune the battery management system for optimal performance

Week 15-16: 4/7 - 4/19

- Research into alternative methods for implementing ackerman drive on the robot.
- Continue working on the electric speed controller
- Start developing and integrating sensor fusion algorithms for better localization

Week 17-18: 4/21 - 4/23

- Conduct full system testing
- Test on the integration with machine learning algorithms
- Document the development process and standards for every module.

B. Software – Jingxuan Xiao

1) Features

- Monocular Depth Estimation
- Object Detection
- 3D Position Estimation
- Pick-up Planning
- Robot Arm Control Integration

2) Tasks

a) Monocular Depth Estimation
   - Research and select model
   - Integrate with camera

b) Object Detection
   - Research and pick a model
   - Integrate with camera

c) 3D Position Estimation
   - Convert 2D data to 3D positions
   - Calibrate with robot’s coordinates

d) Pick-up Planning
   - Develop algorithm
   - Integrate with position estimation

e) Robot Arm Control Integration
   - Interface with arm system
   - Implement movement and grasping

f) System Integration
   - Combine all components
   - Optimize performance

g) Testing and Refinement
   - Develop and run tests
   - Fix issues

3) Milestones

a) Milestone 1: Setup and Initial Prototypes

- Project Setup and Planning (Week 1-2: 10/7 - 10/20)
- Depth Estimation Prototype (Week 3-4: 10/21 - 11/3)

b) Milestone 2: Object Detection and Integration
- Object Detection Prototype (Week 5-6: 11/4 - 11/17)
- Initial Integration (Week 7-8: 11/18 - 12/1)

c) Milestone 3: 3D Positioning and Pick-up Planning
- 3D Position Estimation (Week 9 - 10: 1/13 - 1/25)
- Pick-up Planning Algorithm (Week 11 - 12: 1/27 -2/8)

d) Milestone 4: Robot Arm Integration
- Robot Arm Control Integration (Week 13 - 14: 2/10 - 2/22)
- Pick-up Action Implementation (Week 15 - 16: 2/24 - 3/8)

e) Milestone 5: System Integration and Optimization
- Full System Integration (Week 17 - 18: 3/10 - 3/22)
- Performance Optimization (Week 19 - 20: 3/24 - 4/5)

f) Milestone 6: Testing, Refinement, and Documentation
- Comprehensive Testing (Week 21 - 22: 4/7 - 4/19)
- Final Refinement and Documentation (Week 23: 4/21 - 4/23)

III. Deliverable Artifacts

A. Hardware

1) Description

The deliverable artifact for the first semester will be a robot base equipped with a gyroscope and accelerometer. The robot can be charged with type C cable. Users can navigate the robot with a game controller. It also provides the interface for automatic navigation algorithms to control the robot. For the second semester, an example module will be developed to demonstrate the full capacity of the robot.

2) Documentation

Due to the modular nature of the design, we will provide standardized dimensions or interfaces for every module. Functions in the firmware will be documented and explained in the GitHub wiki. The project will be open sourced on GitHub.

3) Maintenance

The full Rover Master project will be organized in a GitHub organization. We encourage people to add issues and pull requests to improve the product.

B. Software

1) Description

The complete source code for the robotic vision and control system, including modules for monocular depth estimation, object detection, 3D position estimation, pick-up planning, and robot arm control integration.

2) Dissemination Plan

Source code will be on a GitHub repository
Code will be accessible to team members and stakeholders
There will be clear documentation of the code structure for future changes

IV. Mockups

A. Interfaces

The Rover Master features both physical and software interfaces. Physical interfaces include control buttons and sensors such as IMU, optical flow, and Lidar, providing direct interaction with the robot. The software interfaces, including those for machine learning models like monocular depth estimation and object detection, connect the sensors and cameras, forming an integrated user control and monitoring experience.

B. Systems

Rover Master comprises multiple integrated subsystems. The core control system uses an STM32 microcontroller, while the drive system employs brushless DC motors paired with Mecanum or customized wheels. The various components interact via protocols like DShot for motor control and USB for sensor connections. Draft models of each subsystem show the interaction and connectivity between hardware and software.

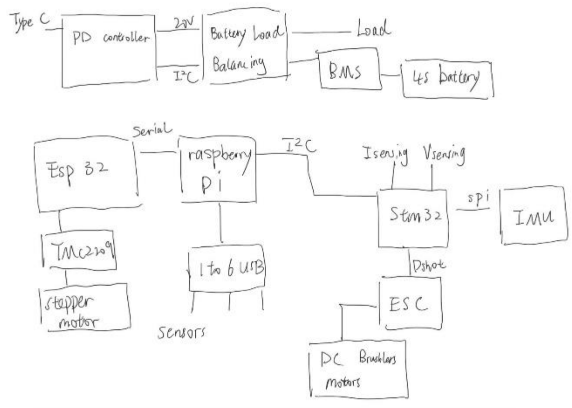
C. Networking

The robot's networking features include communication between the onboard Raspberry Pi (or other computing units) and external devices through Wi-Fi or other networking protocols. Models depict communication across subsystems, emphasizing reliable sensor integration and the coordination needed for real-time control.

D. Storyboards

Storyboards for Rover Master cover different operational modes, including manual control, autonomous navigation, and object pickup sequences. These storyboards illustrate the decision-making process of the robot, from sensor input to motor actions, showcasing how different tasks can be executed seamlessly in real time.

E. Draft Schematics



REFERENCES

[1] Elephant Robotics, “myAGV 2023 Jetson Nano,” Elephant Robotics. Available: https://shop.elephantrobotics.com/products/myagv-2023-jetson-nano?currency=USD&variant=47262874534200&utm_source=google&utm_medium=cpc&utm_campaign=Google+Shopping&stkn=e8fce6917475&com_cvv=8fb3d522dc163aeadb66e08cd7450cbbdddc64c6cf2e8891f6d48747c6d56d2c. [Accessed: Oct. 9, 2024].J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Clearpath Robotics, “TurtleBot 4 Mobile Robot,” RobotShop. Available: https://www.robotshop.com/products/clearpath-robotics-turtlebot-4-mobile-robot. [Accessed: Oct. 9, 2024].K. Elissa, “Title of paper if known,” unpublished.