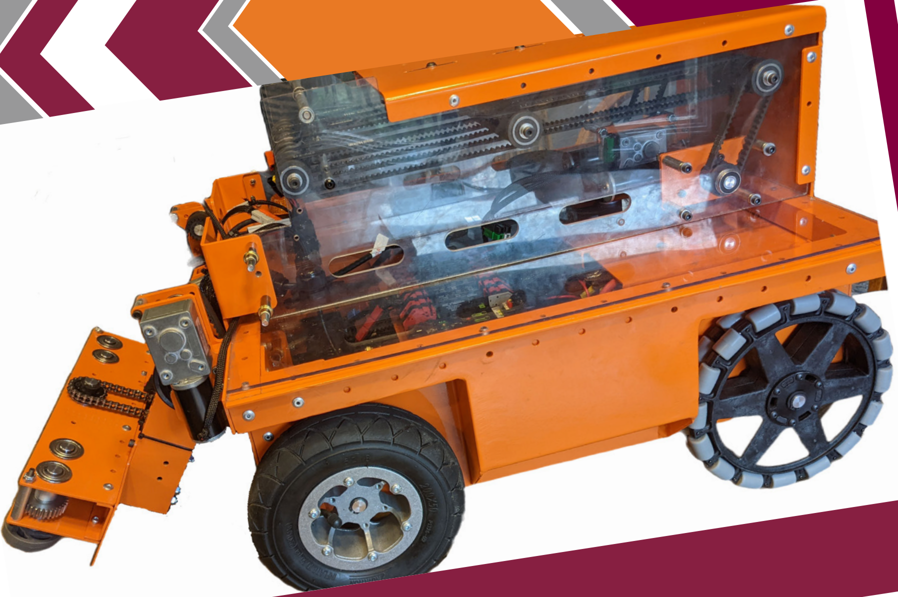
Multi-Incident Response Vehicle

July 2023 | Final Report



SAFE-D
SAFETY THROUGH DISRUPTION

VIRGINIA TECH TRANSPORTATION INSTITUTE
VIRGINIA TECH

Texas A&M Transportation Institute

San Diego State University
Leadership Starts Here

Disclaimer

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. 06-013	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Multi-Incident Response Vehicle		5. Report Date July 2023	
		6. Performing Organization Code:	
7. Author(s) Elijah W. Vaughan John Wiens Michael English Jacob Frye		8. Performing Organization Report No. 06-013	
9. Performing Organization Name and Address: Safe-D National UTC Virginia Tech Transportation Institute 3500 Transportation Research Plaza Blacksburg, VA 24061		10. Work Unit No.	
		11. Contract or Grant No. 69A3551747115/06-013	
12. Sponsoring Agency Name and Address Office of the Secretary of Transportation (OST) U.S. Department of Transportation (US DOT)		13. Type of Report and Period Final Research Report Start Date: 7/2022 End Date: 7/2023	
		14. Sponsoring Agency Code	
15. Supplementary Notes This project was funded by the Safety through Disruption (Safe-D) National University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program.			
16. Abstract The VTTI project team led a collaborative effort with Neaera Consulting Group to develop and integrate a Multi-Incident Response Vehicle (MIRV) into the Safely Operating Automated Driving Systems (SOADS) vehicle. The MIRV vehicle will be applied as one technical solution to how automated driving systems can be designed to interact safely with public safety in challenging scenarios. This project explored whether a MIRV can extend the perception of an ADS to beyond the vehicle by providing eyes on the ground for better situational awareness, deploy flares to secure a scene surrounding a driverless vehicle, and communicate with emergency, safety, and police personnel. The MIRV will be docked underneath the SOADS F-150 vehicle and can be automatically deployed for both autonomous operation and remote teleoperations. Control and supervision of the MIRV will be possible through the cloud, accessible via mobile device and remote fleet management software. The MIRV will potentially demonstrate a new application of autonomous technology in transportation that could greatly improve safety by performing dangerous tasks, allowing passengers to remain safely in the vehicle while it is disposed on the roadway or perform necessary functions outside the vehicle when there are no passengers at all.			
17. Key Words Automation, ADS, Extending, Outside Vehicle, Robot, Remote		18. Distribution Statement No restrictions. This document is available to the public through the Safe-D National UTC website, as well as the following repositories: VTechWorks, The National Transportation Library, The Transportation Library, Volpe National Transportation Systems Center, Federal Highway Administration Research Library, and the National Technical Reports Library.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 16	22. Price $0

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Abstract

The Virginia Tech Transportation Institute (VTTI) project team led a collaborative effort with Neaera Consulting Group to develop and integrate a Multi-Incident Response Vehicle (MIRV) into the Safely Operating Automated Driving Systems (SOADS) vehicle. The MIRV vehicle will be applied as one technical solution to how automated driving systems can be designed to interact safely with public safety in challenging scenarios. This project explored whether a MIRV can extend the perception of an ADS to beyond the vehicle by providing eyes on the ground for better situational awareness, deploy flares to secure a scene surrounding a driverless vehicle, and communicate with emergency, safety, and police personnel. The MIRV will be docked underneath the SOADS F-150 vehicle and can be automatically deployed for both autonomous operation and remote teleoperations. Control and supervision of the MIRV will be possible through the cloud, accessible via mobile device and remote fleet management software. The MIRV will potentially demonstrate a new application of autonomous technology in transportation that could greatly improve safety by performing dangerous tasks, allowing passengers to remain safely in the vehicle while it is disposed on the roadway, or perform necessary functions outside the vehicle when there are no passengers at all.

Acknowledgements

This project was funded by the Safety through Disruption (Safe-D) National University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program. Acknowledgements are made to Jean Paul Talledo Vilela, Senior Technology Implementer, Mike Mollenhauer, Director of the Division of Technology Implementation, Andy Petersen, Chief Engineer, and Neaera Consulting for their contributions to the development and reporting of this project.

Table of Contents

TABLE OF CONTENTS .... **III**

LIST OF FIGURES .... **V**

LIST OF TABLES .... **V**

INTRODUCTION **1**

BACKGROUND **2**

METHOD **2**

MIRV Rover Technical Design Overview **2**
- Software API's for MIRV Management 3
- Cloud Communications 3
- MIRV Operational States 4
- MIRV Fault States 4

Hardware/Software Overview **5**
- MIRV Hardware Design 5
- Garage Hardware Design 9
- Software Design 11

RESULTS **11**

Test Plan **12**

Test Results **13**
- Pi-Lit Deployment 13
- Latency Evaluation 13
- Secondary Goals 13

DISCUSSION **14**

Current Limitations **14**
- Pi-Lit Placement and Control 14
- Pi-Lit Identification 14

CONCLUSIONS AND RECOMMENDATIONS **14**

Areas for Improvement 15

Potential Adaptations 15

ADDITIONAL PRODUCTS 15

Education and Workforce Development Products 15

Technology Transfer Products 16

REFERENCES 17

SUPPORTING ARCHITECTURE CHARTS 18

List of Figures

Figure 1. Photo. Summation of Pi-Lit non-incendiary road flares 2

Figure 2. Illustration. Rendering of MIRV. 6

Figure 3. Graph. Rover chassis gear ratio trade-offs motion profiling comparison. 6

Figure 4. Graph. Rover chassis gear ratio movement characteristics. 7

Figure 5. Illustration. Garage dynamic section. 9

Figure 6. Illustration. Garage static section. 9

Figure 7. Diagram. Garage safety lock. 10

Figure 8. Photo. Spare tire area to be removed, view from underneath. 10

Figure 9. Illustration. MIRV parked in collapsed garage. 10

Figure 10. Diagram. Emergency signal placement for commercial vehicle on a divided highway 12

Figure 11. Diagrams. Pi-Lit configurations: top left, five Pi-Lit left taper; top middle, five Pi-Lit right taper; top right, seven Pi-Lit spear; bottom left, commercial distance three Pi-Lit t 13

Figure 12. Flowchart. MIRV normal operational states. 18

Figure 13. Flowchart. Robot cloud communications. 18

Figure 14. Flowchart. Software architecture. 19

List of Tables

Table 1. Project Assumptions and Limitations 3

Table 2. MIRV State List 4

Table 3. MIRV Fault State List 4

Introduction

The Multi-Incident Response Vehicle (MIRV) is intended as one part of the larger Safe Operating Automated Driving Systems (ADS) in Challenging Dynamic Scenarios (SOADS) project. SOADS is facilitating the development of ADS-equipped vehicles that can safely interact with public service and infrastructure providers. This collaboration brings together some of the largest national voices within the OEM, industry owners and operators, and public safety communities to develop solutions. The project team is working extensively with stakeholders to identify targeted dynamic scenarios and define the interactions between the actors and ADS, as well as develop technological solutions to address each scenario. A Ford F-150 will be equipped with a prototype open-reference Level 4 plus (L4+) ADS package that includes the ability to exchange data through connected vehicle systems, allowing for cooperation with roadside units. A series of demonstrations will occur showing how the equipped ADS can safely interact with scenarios while operating on corridors optimized for automation. The team will collect comprehensive data to be shared with developers and analyzed with respect to the effectiveness of ADS implementations across the scenarios tested.

VTTI has sourced the talents of Neaera Consulting Group (Neaera), specializing in the field of connected/automated vehicle technologies and architecture, for the creation and delivery of an autonomous MIRV rover intended for use in some of the established SOADS scenarios.

The project's primary goals were to accomplish the following:

1) Autonomously deploy and recover the MIRV from the SOADS F-150 truck.
2) Autonomously deploy and recover non-incendiary LED road flares in spear or taper format based on the SOADS parked location.
3) Develop an eyes-on-the-ground solution via mobile application which will allow users to remotely pilot the MIRV and stream video and sensor data back to the user. The SOADS Robotic Operating System (ROS) will also be able to access the video and sensor streams from the MIRV platform.

The project's secondary goals were to accomplish the following:

1) Interact with on-road personnel by approaching and facilitating two-way audio communication back to a remote user.
2) Act as a mobile warning beacon through an omnidirectional amber LED light atop the MIRV while deploying the non-incendiary LED road flares.

Background

The MIRV addressed the following research questions:

1) Can a mobile rover unit be deployed to provide additional ADS capabilities outside the vehicle in challenging dynamic scenarios and interactions with public safety services?
2) Can a mobile rover unit be deployed to enhance the capabilities of remote fleet management services?

Method

MIRV Rover Technical Design Overview

The MIRV rover was designed to be capable of varying levels of autonomy ranging from near complete autonomy to complete remote teleoperation. In autonomous mode, the robot receives broad directives from a human operator to execute autonomously. Autonomous directives consist of actions such as *"Place Flares"* or *"Dock with F-150."* The non-incendiary flares that MIRV places are branded *Pi-Lits,* which are shown in Figure 1.

Figure 1. Photo. Summation of Pi-Lit non-incendiary road flares (firsttorescue.com, 2023).

Directives are sent to the robot from an Android tablet/cell phone running a unique application developed by Neaera. The Android app can also grant the user full remote operation. This enables the user to have direct access to subsystems such as the drivetrain, the Pi-Lit deployment and retrieval mechanisms, the robot docking system, robot sensor data, and system-wide statistics (e.g., battery life and sensor feedback).

The assumptions and limitations that refined the project scope are listed in Table 1.

Table 1. Project Assumptions and Limitations

#	Assumption/Limitation	Description
1	Research Project	Neaera assumes this is a research project, not a final product deployment. The rover will be designed to operate autonomously in a controlled environment with human supervision.
2	Clear Road and Weather Conditions	The rover will be operated in daylight, clear weather conditions (no snow/rain), on a clean road with well-defined road markings.
3	GPS/LTE Availability	The robot is assumed to be operating in an open sky highway environment where GPS and VTTI Virginia Connected Corridors (VCC) LTE are both available.
4	High-definition (HD) Road Map Availability	The rover assumes that it will have access to HD road maps of the operating site. This can be either through the LTE connection with the internet or by downloading the necessary maps from the drop-off vehicle. Neaera can build these maps for the selected test area if they are not available.
5	VCC Cloud	The rover will perform all cloud operations through the VCC cloud.
6	Android App Deployment	The MIRV user control app will be written in Flutter and will be published to the Android App Store.
7	User Intervention	A user will be available to take over remote operation in the event of a fault.
8	MIRV Deployment and Docking	The MIRV will always be deployed and docked in an environment that is level, solid, and free of obstructions. The rover will dock and undock only when stationary. Validation of proper docking will occur before the host vehicle begins motion. The MIRV will not charge in the dock. The MIRV will be charged before the test.
9	Non-incendiary LED Flares	Pi-Lit flares will be charged and manually loaded into the MIRV rover before operation. The MIRV rover will not be required to charge flares. The MIRV rover will not be required to synchronize flares onboard the rover (they will be loaded in the order to be synchronized).
10	Human Reset	The MIRV rover assumes that humans will be available in between multiple deployment periods to perform actions such as changing the batteries, charging, and reloading LED flares and other rover maintenance.
11	Operation of SOADS vehicle	While the garage is installed and MIRV is docked, the SOADS vehicle will only conduct low-speed maneuvers and not travel on public roadways

Software API's for MIRV Management

The MIRV has a software application programming interface (API) that allows for control of the rover. This API is used to integrate the Android app that functions as the user interface, as well as allow the SOADS F-150 to send commands to the rover. The API will utilize pre-shared static keys for a basic level of authentication and security. Using the API, users will be able to access sensor data and retrieve the MIRV rover's current state. Additionally, the API will be used for sending command directives to the rover and helping to correct any fault states that may occur during operation.

Cloud Communications

For the MIRV, Neaera integrated cloud operation through the VCC cloud infrastructure, providing benefits such as security and interconnectivity with existing VCC resources. The rover communicates with the cloud to upload data to users, communicate with the rover deployment mechanism (garage), and ultimately interact with the SOADS F-150 ADS. This will enable the F-150 to send/receive commands and send/receive data to the rover. The communication is accomplished through Rest APIs, with data/video streaming being done using WebRTC through Ant Media Server. Security is maintained using pre-shared keys for all communications.

MIRV Operational States

Table 2 lists a number of the operational states that the MIRV may put itself in, or that it can be put into.

Table 2. MIRV State List

#	Operational State	Auto Trigger	MIRV Condition
1	Remote Operation	When an authorized user requests control over the rover or if specific faults are hit	The MIRV subsystems are under control of the user. Non-fatal faults are warnings in this mode.
2	Autonomous	Default operation: robot is using default tasks or user-queued macros	The MIRV is actively and autonomously executing tasks using needed subsystems.
3	Idle	When the task queue is empty	All systems are functional and waiting for task.
4	Low Power	When the low-power fault is activated or if it has remained in an idle state for a user-specified time	Shuts off all nonessential processes and waits for command.
5	Remote Observation	When an authorized user requests rover status	This state can occur in addition to another state. The user will get rover status in near real time.
6	Docked	When the robot is on the docking station and has not received a deploy command	The rover is secured into the truck and is put into a low-power state.

MIRV Fault States

Table 3 lists a number of the MIRV's fault states, which prevent damage to itself, other vehicles, infrastructure, and personnel.

Table 3. MIRV Fault State List

#	Fault State	Cause	MIRV Condition
1	Emergency stop	Either the user interface or on-robot button triggered	All robot operations are halted, an authorized user must reactivate through user interface, and on-robot button must be manually reset.
2	Untraversable terrain	The robot is unable to continue its path to a target due to it determining that there is no clear or safe path	The robot is put into remote operation idle mode.

#	Fault State	Cause	MIRV Condition
3	Low power	If the robot detects a battery voltage below a specified value	The rover enters a low-power state.
4	Rover stuck	If the rover detects that is unable to move after attempts to recover itself	The robot is put into remote operation idle mode.
5	Rover damaged	If the rover detects that a subsystem is behaving differently than intended	The robot is put into remote operation idle mode.
6	Deploy attempted in invalid environment	If a deployment is attempted into an environment where it is outside of operating parameters	The robot remains in its dock but is put into remote operation idle mode.
7	Dock attempted in invalid environment	If the rover is requested to dock in a situation where it is unable to safely access its dock	The robot is put into remote operation idle mode.
8	LED flare recovery unsuccessful	If in attempting to recover an LED flare the rover is unable locate or manipulate an LED flare	The rover will notify the user of the status of the failure and continue its task.
9	Missing resource detected	If the rover detects that is unable to access a resource required for one or more of its mission objectives	The rover will notify the user initially and will notify the user if a mission objective is attempted requiring that resource. The mission objective will then be skipped unless that resource is detected or overridden by the user.
10	Unsafe environment detected	If the environment in which the rover is asked to go into is deemed unsafe by its compute unit	The rover will notify the user of the nature of the environment and will attempt to leave the area. It will then be put into the remote operation state.
11	Compute error	If the rover's compute unit encounters a fatal error in one of its processes	The rover will attempt to notify the user, then attempt to restart in the remote operation mode.
12	Loss of comms	If the rover detects that it is no longer able to communicate with its cloud environment for a duration of time	The robot will pause all tasks and will attempt to reconnect until it is successful.

Hardware/Software Overview

MIRV Hardware Design

Design of the MIRV, the flare deployment mechanism, and the garage was conducted by Neaera with close consultation by VTTI. The rover is durable and weather resistant when stored but is intended to operate in dry conditions on paved surfaces. It has been designed to operate within its operational research environment with no barriers to incremental improvements. Two MIRV units were constructed: one a developmental unit kept by Neaera and one production unit delivered to VTTI along with the sole docking garage. Figure 2 shows a rendering of the MIRV.

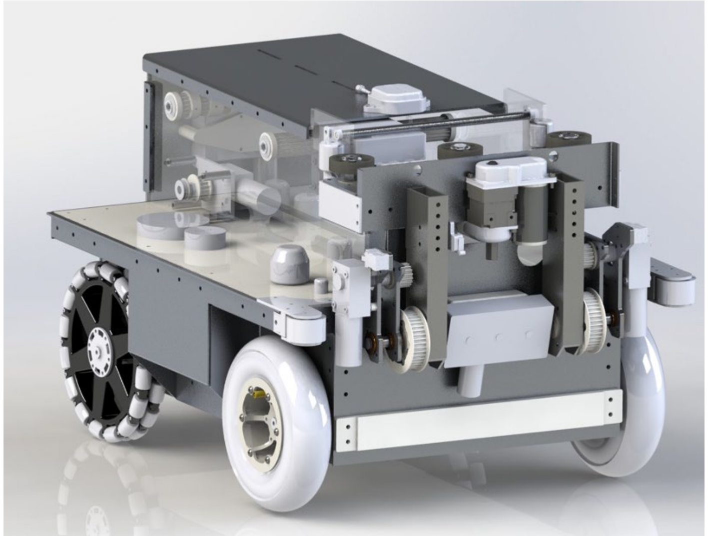


Figure 2. Illustration. Rendering of MIRV.

Chassis Design

The MIRV chassis is designed to be capable of traversing concrete curbs and other debris up to 3 inches in height. With a focus on durability, precision, and simplicity, a drivetrain consisting of two pneumatic wheels at the front and two omnidirectional wheels at the rear was chosen. The pneumatic wheels allow for increased traction for climbing obstacles and the rear omnidirectional wheels allow for precise turns. Using a drivetrain system like this shifts the center of rotation to the front of the robot, allowing for very fine-tuned movements when picking up Pi-Lits as well as docking with the garage. The gear ratio selection for the chassis was selected by comparing parameters such as acceleration, top speed, and power usage. These three factors combine to create a sweet spot in the target wheel speed, allowing for an optimal design. A design gear ratio of 12:1 was decided upon based upon that optimized acceleration and low-speed control while still maintaining a top speed of 15 ft/s.

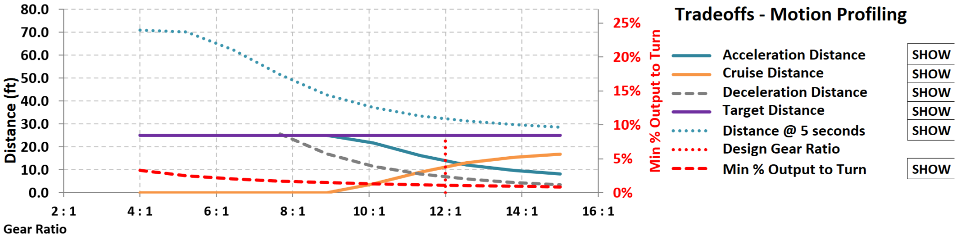


Figure 3. Graph. Rover chassis gear ratio trade-offs motion profiling comparison.

Figure 3 is a comparative motion profile of the chassis with gear ratios ranging from 4:1 to 15:1. An important targeted parameter for the chassis design is optimizing acceleration distance as well as maximizing the top speed of the rover. The acceleration distance in Figure 3 should be minimized, meaning that it takes less distance to reach the maximum speed while the cruise distance is maximum. The best gear ratio for these goals is 12:1, as this will optimize the goal parameters of acceleration, top speed, and power usage.

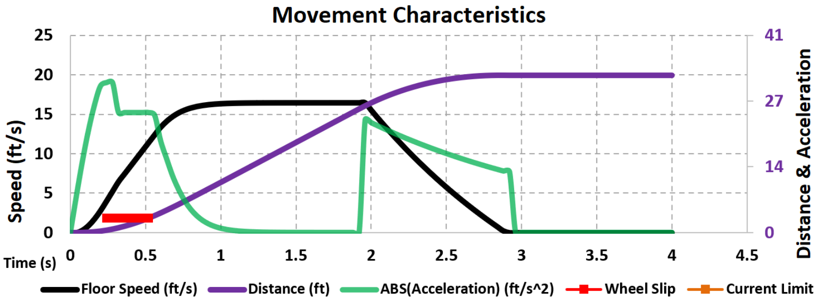


Figure 4. Graph. Rover chassis gear ratio movement characteristics.

Figure 4 illustrates the performance characteristics of the drivetrain with the given parameters and goals. This graph displays the rover's kinematic performance over a 25-ft long sprint with the motors set to brake mode. As shown in the simulated plot, near the beginning of the sprint the rover is traction limited, meaning that the wheels' friction is the limiting factor to acceleration. This means that any increase in gear ratio will do nothing to help with acceleration. Another key point from this diagram is that the maximum velocity of the system is reached in under a second, leading to a very responsive system.

Structure

The chassis structure consists of a rigid 6061 aluminum tube frame to provide both strength and structural rigidity at light weight. All subsystems mount to this frame, and it serves as a basis for interfacing between systems. An outer layer of 5052 sheet metal interfaces directly to the frame and strengthens all the tube joints in addition to sealing the rover. All the sheet metal on the chassis is 0.09-in. thick to reduce weight and all the seams are glued together with a hybrid polymer-based sealant. To further reduce water ingress, all pop rivets on the chassis section of the MIRV are sealed rivets. The bottom panel of the chassis is made from UHMW plastic and attached to the frame with 12 ¼-20 fasteners. This panel serves as the most convenient method to access the inside of the rover, as the top panel requires both the conveyor and intake subassemblies to be disassembled. The battery is removable from the rear of the rover and requires four ¼-20 fasteners to be removed before the battery compartment is accessible.

Intake

The Pi-Lit intake for the MIRV is designed to be capable of reorienting incoming Pi-Lits as well as centering the Pi-Lit for the corresponding conveyor side. The intake is capable of being in two states, a deployed state and a stowed state. During normal operation, the intake will be in the stowed state to allow for increased ground clearance. The rotation of the intake arm is driven by two worm-driven gearboxes that are geared for 65 RPM. These gearboxes are not capable of being back driven and must be driven by the motors to move between states. The deployed and stowed states are defined by the limit switches mounted at the base of the arm. In both states there are hard stops embedded into the limit switch mounts to take any external loads to prevent overextension.

At the base of the arm, the Pi-Lit intake wheels are positioned to align incoming Pi-Lits into one of two channels. These channels align with the Pi-Lit conveyor system to allow for easy transfer between the two subsystems. The intake wheels are designed to be easily compressible, allowing for the wheels to conform to incoming shapes and help in reorienting any incoming Pi-Lits.

Conveyor

The Pi-Lit conveyor is capable of storing up to eight Pi-Lits with four Pi-Lits being in each channel of the conveyor. Each side of the conveyor has two sets of timing belts that are driven separately by worm-driven gearboxes geared to 160 RPM. The timing belts create a surface that acts as a conveyor for Pi-Lits, allowing for the Pi-Lits to be transported linearly across the length of the rover. At the entry point of the conveyor, there are several rollers on all sides to prevent the Pi-Lits from jamming on entry.

Wired Network

The AGX driving the MIRV rover is equipped with four POE ethernet ports that are used for network communication around the rover. These ports are labelled 1-4 on the AGX, with each port mapping to the corresponding network adapters eth1–eth4 on the AGX. The eth1 ethernet port is left unconnected to any device and instead runs from the AGX to the battery bay via an ethernet extension cable. This port is intentionally made available for the user to configure as they see fit. By default, this port is set up to use DHCP for IP resolution and will accept an IP assigned to it by a connected router. This is useful for connecting the MIRV via ethernet to the backend network for network-intensive tasks such as backing up the AGX or downloading large packages. This port can also be used for installing additional POE hardware if needed.

Wireless network

The MIRV rover is equipped with an RUT 240 LTE/Wi-Fi router. This router broadcasts a 2.4-GHz Wi-Fi network called MIRV_WIFI, which can be connected to or from nearby devices. This network supports a multitude of use cases necessary for the MIRV rover to effectively operate, including:

1) Providing a network connection for the garage to connect to the cloud.
2) Providing a network connection for a nearby tablet to connect to the cloud.
3) Providing an easy way for developers to connect to and monitor the MIRV rover.

Note, the internet connection provided by the LTE router connects directly into the VTTI network. This allows the rover to connect to online resources internal to the VTTI network such as the MIRV API and GitLab, but it is not accessible via the public internet for security.

Controller Area Network

The MIRV relies on an internal Controller Area Network (CAN) bus network that is used to communicate with the motor controllers and power distribution module (PDP). This network is running a 1-MHz CAN 2.0 FD network. This CAN network communicates with the AGX via a USB to CAN adapter plugged into the top USB port on the front of the AGX. Despite having two integrated CAN ports (can1, can2) for working with CAN, the AGX needs the USB 2 CAN adapter in order to hit the very precise CAN timings required by the CTRE motor controllers to talk to the CAN effectively.

Garage Hardware Design

The garage mechanical system functions by having a stiff static section that mounts directly to the F-150 as well as a dynamic section that is driven upwards/downwards by position-controlled linear actuators. The fixed structure directly mounts to the frame of the F-150 via three U-bolts secured directly to the 2-in. × 1.5-in. cross member tubes. Figure 5 and Figure 6 illustrate the dynamic section and the static section. These two sections are joined by 1-in. OD aluminum tubes that slide on linear slide bearings with a linear actuator mounted center on each side. The MIRV rover can align with the deployed garage by referencing the mounted ArUco tags for side-to-side alignment as well as inductive proximity sensors and an infrared beam break to detect when the rover is in the garage. After the garage detects the rover, it will transition from the deployed state to a stowed state where the rover is secured by compressing a 6-in. section of foam to prevent movement while the vehicle is moving.

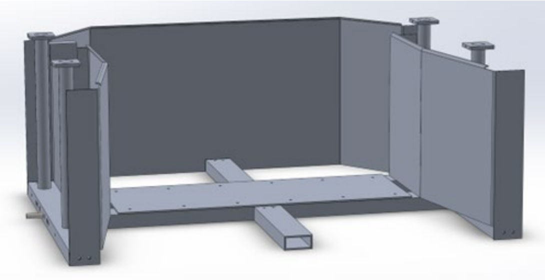


Figure 5. Illustration. Garage dynamic section.

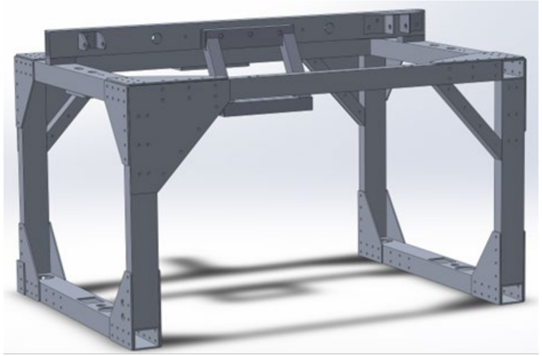


Figure 6. Illustration. Garage static section.

Additionally, the MIRV garage hardware system is actuated by two separate IP66 sealed linear actuators. These actuators are each capable of providing 400 lb of force while providing Hall effect position feedback to the controller. The garage assembly has integrated limit switches that trigger at the upper and lower states and can be adjusted by rotating the threaded mount. The linear actuators are controlled with a current-limiting factor to prevent the motors from applying too much force. Furthermore, an over-center linkage is included with the garage system to serve as a safety measure to ensure that the dynamic section of the garage cannot deploy when the garage is in the stowed state. Figure 7 shows the garage lock in the locked state.

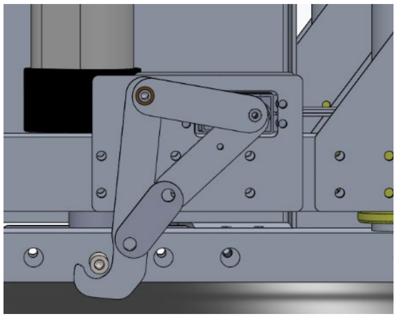


Figure 7. Diagram. Garage safety lock.

Garage Installation

The MIRV garage is designed to interface with the provided 2022 F-150 Lariat in the spare tire area. The garage mounts to the F-150 Lariat by using three U-bolts that secure the static portion of the garage to the crossbeams of the F-150. Prior to installation, the F-150 must be modified to accommodate the space requirements of the garage. The modification requires the removal of the panel that is above the spare tire when it is stored as shown in Figure 8 and Figure 9.

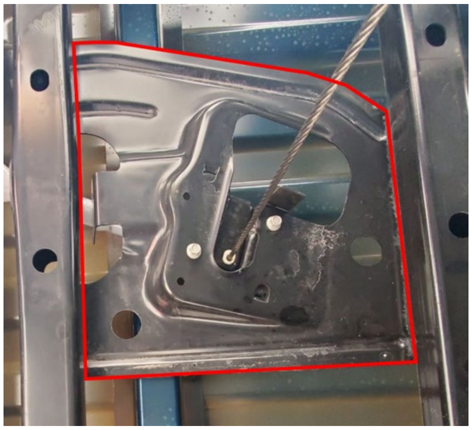


Figure 8. Photo. Spare tire area to be removed, view from underneath.

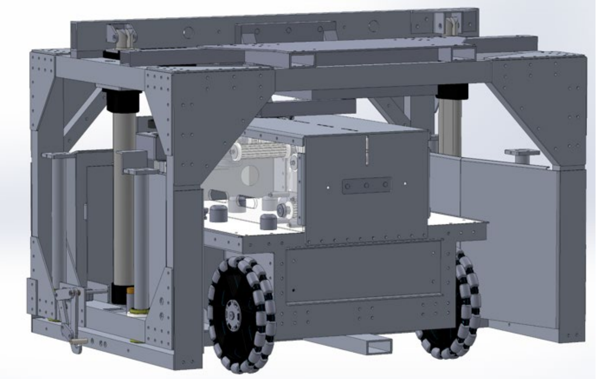


Figure 9. Illustration. MIRV parked in collapsed garage.

ArUco Tag Localization

The garage is detected and located by three ArUco tags. These tags were printed 6-in. square on 8-in. square sheets of synthetic paper. The three tags used were placed on the back and sides of the garage. This gives the MIRV the ability to see the tags from just about anywhere, with a tag at the back for the final drive in, and commonly allowed the MIRV to see more than one tag at once. For each tag, the cv2 ArUco library is used to detect and estimate the tag's pose. The rover's position is calculated relative to each marker very accurately (within a few inches). This allows the rover to set itself up directly in front of the garage before driving in and docking to minimize the chance of the rover hitting the edge of the garage.

Software Design

The MIRV rover uses a variety of sensors to gain a comprehensive view of its environment such as cameras, encoders, GPS, and inertial measurement units (IMU). Additionally, the robot may pull additional environment information such as HD maps from the SOADS vehicle. All sensor data is processed onboard the rover and used for navigation and motion planning. The rover records available sensors locally for future use in training or simulation. A smaller selection of data feeds is also sent to the cloud for real-time monitoring and control purposes. A full breakdown of the software architecture can be found in the appendix.

ROS Architecture

The MIRV rover code is broken down into two ROS packages: mirv_real and mirv_control. The mirv_real package contains the code necessary to interface with physical hardware such as the motor controllers or cameras. The mirv_control package is used for handling the logic and control flow of the rover and includes components such as path planning and handling the connection to the cloud. The majority of node communication is done in the control package. These two ROS packages are designed to be run on Ubuntu 18.04 using ROS Melodic.

In addition to the two packages mentioned above, the MIRV also takes advantage of multiple community-distributed packages for operating the rover, such as the Ublox_f9p and actionlib libraries.

Key cloak

The MIRV system takes advantage of VTTI's existing Keycloak server to handle authentication for the MIRV rover. All calls needed by the cloud to connect to the MIRV rover must first be authenticated by Keycloak to ensure a secure connection. The initial connection to Keycloak is done using username and password authentication from the rover. Once the provided credentials are validated, Keycloak generates a token that is used for all further communications.

Results

Test Plan

The MIRV was expected to demonstrate repeatable success at the target goals listed in the *Introduction* under the assumptions listed in the *Methods* section. This demonstration was to be conducted at VTTI. While the SOADS vehicle was not available at the time of delivery, all demonstrations were to be conducted as if the vehicle was present. The goals are additionally listed below with further detail:

1) Autonomously deploy and recover the rover from a garage, representative of the enclosure that will be integrated into the SOADS vehicle.

2) Autonomously deploy and recover 10 (minimum requirement 5) non-incendiary LED flares in spear, taper left, or taper right format based on a simulated parked location. This will be required to take no more than 10 minutes from the command being issued to the last flare being place with a goal of under 3 minutes. Placement of non-incendiary flares must adhere to, at minimum, the placement described in 49 CFR § 392.2 - Emergency signals; stopped commercial motor vehicles Paragraph (b) for a divided highway such as in Figure 10.

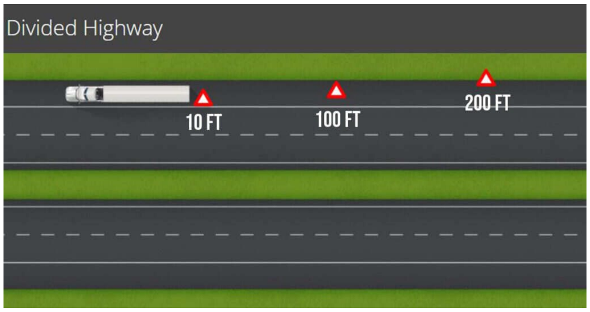


Figure 10. Diagram. Emergency signal placement for commercial vehicle on a divided highway (papertransport.com, 2023).

3) Function as an eyes-on-the-ground solution via mobile application that will allow users to remotely pilot the robot and stream video and sensor data back to the user. The latency of this remote operation will be required to be under 500 ms with a goal of under 350 ms. The streamed sensor data will include:

a. Current GPS coordinates
b. Camera feed
c. Heading information
d. Robot status information (which systems are working on the robot, red/yellow/green status lights for different subsystems and sensors)
e. Status of Pi-Lits (how many have been deployed, what is their current state)
f. Sensor fused map overlay (This will be a two-dimensional map that shows what the robot thinks of its environment. It will include things like the location of detected lane lines, where the Pi-Lits are, where the garage is, and the location of any other obstacles that the rover cannot pass.)

Test Results

Pi-Lit Deployment

The MIRV successfully completed all required tasks within the established requirements. Upon final delivery and handover to VTTI, the MIRV repeatedly deployed from the garage and redocked, autonomously laid and retrieved Pi-Lits in lane-taper and spear formations, and functioned as an eyes-on-the-ground solution with appropriate latency. The formations for the Pi-Lits that were achieved are shown in Figure 11. All non-commercial formations had 10-ft longitudinal spacing and equal lateral lane spacing between Pi-Lits; commercial spacing requires 100-ft longitudinal separation.

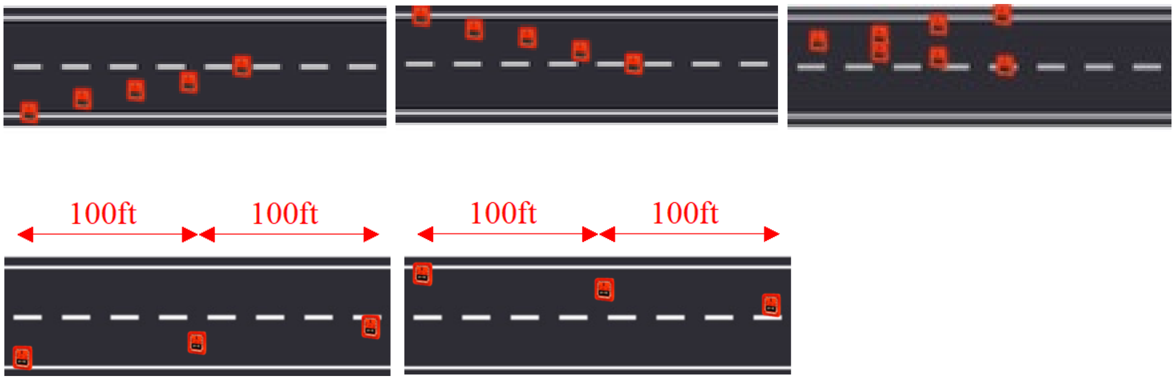


Figure 11. Diagrams. Pi-Lit configurations: top left, five Pi-Lit left taper; top middle, five Pi-Lit right taper; top right, seven Pi-Lit spear; bottom left, commercial distance three Pi-Lit taper left; bottom middle, commercial distance three Pi-Lit taper right.

Videos of the MIRV accomplishing these tasks will be uploaded to the SAFE-D dataverse at https://dataverse.vtti.vt.edu/dataverse/safed/. All formations were laid in under 10 minutes. During testing, the MIRV successfully deployed, laid a seven Pi-Lit spear, retrieved all Pi-Lits, and redocked in under the 10-minute time limit. Most formations took under 5 minutes to deploy.

Latency Evaluation

The video stream latency of the MIRV was evaluated by showing the MIRV a video of a clock and streaming the clock time over the network to the tablet. A picture was then taken of both the tablet and the original time. The streaming latency could then be evaluated by comparing the time shown on the original laptop screen with the time shown on the tablet screen. Using this method, it was determined that the system has approximately 400 ms of latency in streaming from the MIRV to the tablet when being controlled on the Android tablet. Motor control had much lower latency, virtually imperceivable.

Secondary Goals

The MIRV partially achieved the secondary goal of functioning as a mobile warning beacon by utilizing an omnidirectional light atop its housing; however, the light can only be seen from the front, rear, and right side as it is blocked by the Pi-Lit conveyor from the left side. Due to timeline and budget constraints, the secondary goal of facilitating two-way communication was not met, though the design does enable this to be easily implemented in the future.

Discussion

Current Limitations

Pi-Lit Placement and Control

Currently, the MIRV can hold up to eight Pi-Lits, though it is only programmed to make formations of up to seven. All of these Pi-Lits can be controlled via the MIRV interface with a controller provided from the manufacturer. The Pi-Lits can be commanded to flash in unison or in sequence, though the Pi-Lits must be turned on and loaded into the MIRV in the order that they are to be laid. To overcome this limitation, further negotiation is required with the manufacturer.

Pi-Lit Identification

Currently, Pi-Lit detection utilizes a custom-trained ResNet-50 neural network, running on the AGX. This network occasionally fails to detect Pi-Lits. Issues with the neural network are described below:

1. Detections can commonly fail (up to 50%) under the following conditions:
    a. Pi-Lit is on or very near a lane line.
2. Detections can rarely fail under the following conditions:
    a. Pi-Lit is partially shadowed.
    b. Pi-Lit shadow is significant (late in the day).
    c. Camera is pointed directly at the sun.
3. False positives are possible. These are rare but can occur somewhat frequently with orange/brown rocks/dirt on the road surface. To reduce the rate of false positives, the following filters have been placed on detections:
    a. Must be at least 30 pixels wide and 8 pixels tall.
    b. Cannot touch the edge of the frame (top, bottom, left, right).

The frame rate of the network is around 2 fps. Because of this low data rate, live feedback is not present when navigating towards a Pi-Lit.

Conclusions and Recommendations

As shown by the demonstration of MIRV's capabilities at VTTI's Smart Roads, this system is a great first step and an excellent proof of concept that the reach of a driverless vehicle system and ADS capabilities can be extended beyond the confines of the vehicle itself. This technology has the potential to eliminate the need for passengers to exit their vehicles onto a live roadway, greatly improving safety for pedestrians and other road users. The MIRV also offers a glance into what a deployable robot could do to broaden a remote fleet management operator's understanding of a situation by extending perception beyond the typical vehicle-mounted sensor suite when a driverless vehicle is disposed.

Areas for Improvement

As the MIRV is currently configured, the height of its camera sometimes prevents adequate lane mapping, especially when lines are broken and spaced at the standard for most limited access highways or if the lines are degraded/faded. Test photographs were taken at varying heights (6 in., 1 ft, 2 ft, 3 ft) of a variety of different lane configurations (two solid lines, one dashed line, two dashed lines) and run through the current neural network on the MIRV used to map its environment. Improvement was seen at all increased heights, though the most significant jump in improvement was seen at 2 ft of additional height. As a potential improvement to MIRV, its camera could be placed on an actuated lever to assume this height increase and realize the benefits with minimal changes needed to the hardware platform or software.

Potential Adaptations

One key potential area in transportation safety for the MIRV to be greatly beneficial would be in the realm of commercial trucks. The MIRV is currently only designed to handle flare placement for a commercial vehicle when traffic is approaching from one direction. It is not designed to circumnavigate the vehicle it is deployed from and deploy flares in front of the vehicle, as is required for a commercial vehicle pulled over on a non-divided highway. This technology would allow the driver to remain in the vehicle while flares are deployed, greatly reducing risk to the driver, or enable a driverless system to comply with the 10-minute flare placement requirement for commercial vehicles. In order to accomplish this, the MIRV would need to be redesigned with a lower overall height to pass underneath the deploying vehicle or be able to navigate a path safely around the deploying vehicle without obstructing traffic.

Additional Products

The Education and Workforce Development (EWD) and Technology Transfer (T2) products created as part of this project are described below and are listed on the Safe-D website here.

Education and Workforce Development Products

Students, both high school and college, played a major role in the design, development, and testing of the MIRV. All students were supervised, and their work was reviewed by Neaera and VTTI engineers. Six high schoolers from the Highlanders Robotics Team in Fort Collins, Colorado, and two collegiate students from Colorado State University were sourced as interns. All of these students come from science, technology, engineering, and math (STEM) backgrounds, and this project served to further enrich their skills and aptitude in the realm of robotics and autonomy while also granting them exposure to design around automotive platforms and the role that engineers play in road user safety. The students have all provided a brief synopsis of their experience and lessons learned, which will be submitted to SAFE-D.

Technology Transfer Products

Four major items were delivered at the conclusion of the project:

1) A fully functional MIRV and garage for storage and deployment of the MIRV, integrable with the F-150 SOADS vehicle.
2) A complete software package of operational source code for controlling and communicating with the MIRV.
3) Exhaustive demonstration of functionality and requirements validation conducted at the Smart Roads along with data collected from these tests (video, requirement validation results, various operational data, ROS bags, etc.).
4) A final report discussing how the research questions were addressed and the results of the demonstration.

References

FirstOut. (n.d.). *Sequential Smart LED Flare – Alkaline*. https://www.firstoutrescue.com/sequential-smart-led-flare-alkaline.html. Accessed March 29, 2023.

Paper Transport. (n.d.). Emergency Triangle Placement. https://www.papertransport.com/blog/emergency-triangle-placement. Accessed March 29, 2023.

Appendix

Supporting Architecture Charts

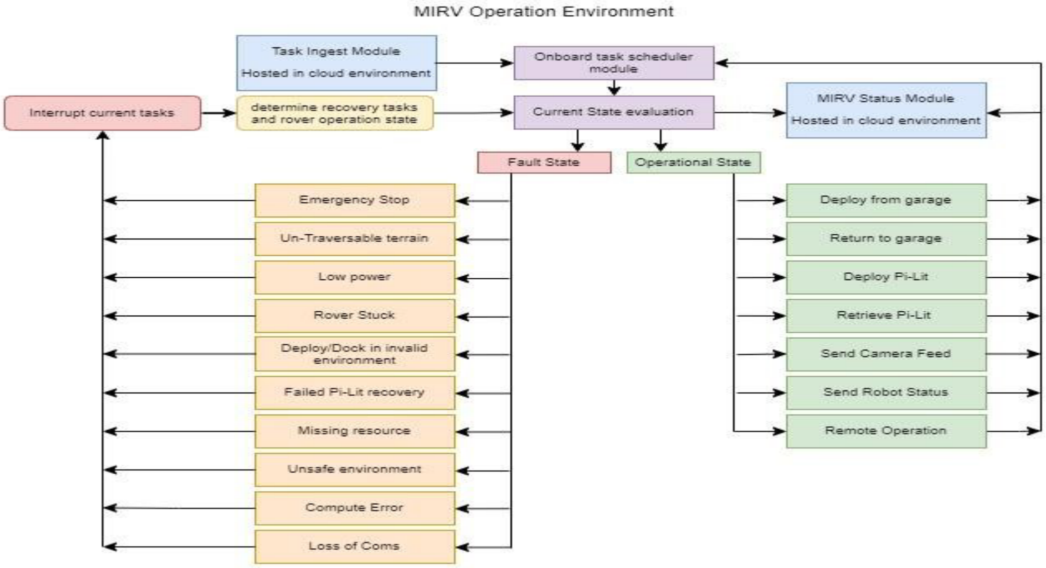
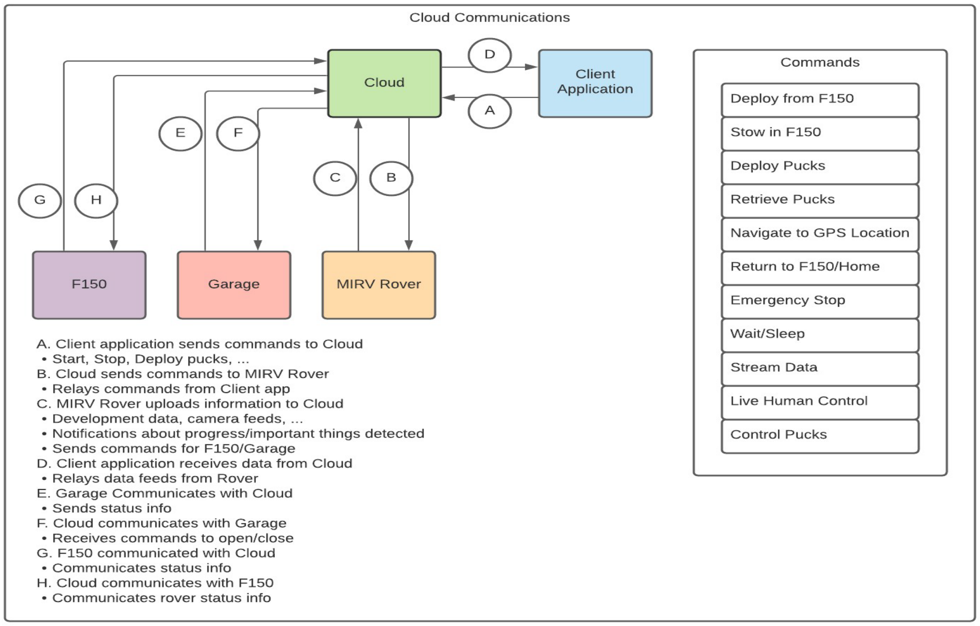


Figure 12. Flowchart. MIRV normal operational states.



A. Client application sends commands to Cloud
- Start, Stop, Deploy pucks, ...

B. Cloud sends commands to MIRV Rover
- Relays commands from Client app

C. MIRV Rover uploads information to Cloud
- Development data, camera feeds, ...
- Notifications about progress/important things detected
- Sends commands for F150/Garage

D. Client application receives data from Cloud
- Relays data feeds from Rover

E. Garage Communicates with Cloud
- Sends status info

F. Cloud communicates with Garage
- Receives commands to open/close

G. F150 communicated with Cloud
- Communicates status info

H. Cloud communicates with F150
- Communicates rover status info

Figure 13. Flowchart. Robot cloud communications.

The MIRV codebase is structured such that there is a logical flow of information from inputs to the system, such as encoders, GPS, and cameras, to outputs from the rover, such as cloud updates and the physical motion of the rover. This is depicted in the code block architecture below.

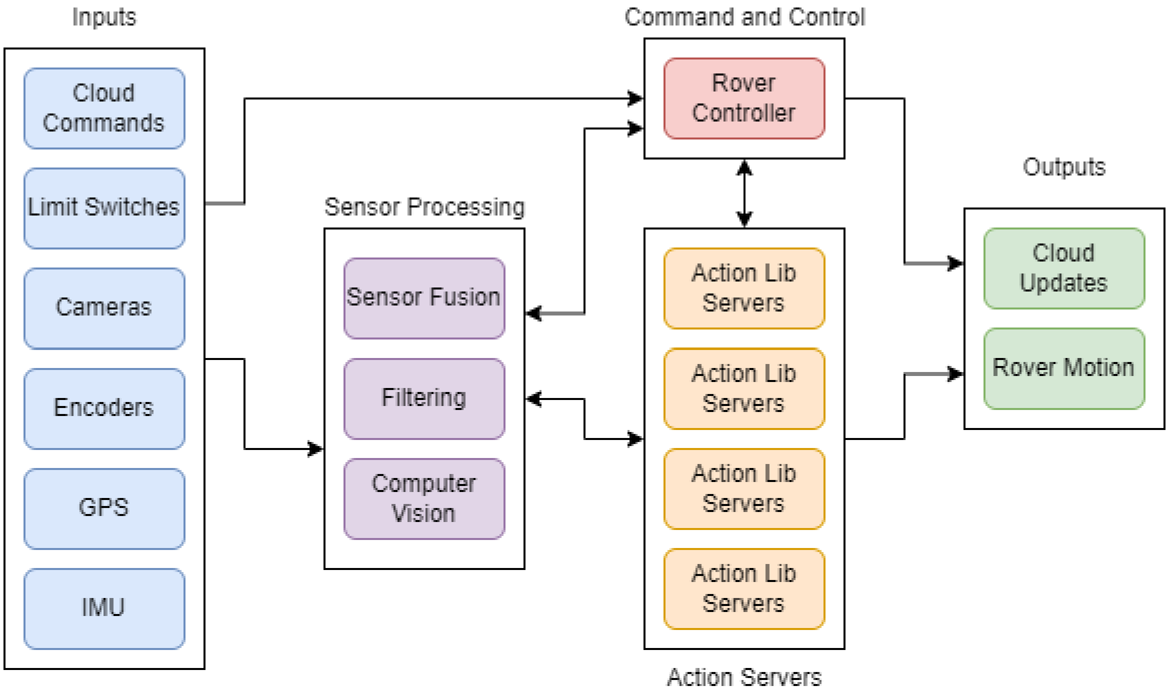


Figure 14. Flowchart. Software architecture.

Starting from the left, information about the state of the rover is ingested from a variety of sensors, including inputs from both physical sensors on the rover as well as remote inputs from either the cloud or a connected client application (e.g., the tablet). These remote inputs are collectively notated as "Cloud Commands." Information from these two sources is then passed on to "Sensor Processing" and "Command and Control," respectively. The "Command and Control" block of the rover code is responsible for handling what the rover will do next, including maintaining the rover's state and performing verification to ensure new actions can be performed effectively given the rover's present state and configuration. The "Sensor Processing" block of actions is responsible for extracting meaningful conclusions about the state of the rover and its world given the raw inputs supplied by devices such as cameras. This includes tasks such as filtering noisy inputs; converting camera frames into useful data about the locations of the Pi-Lits, lane-lines, and ArUco markers; and fusing the IMU, wheel encoders, and GPS into a common representation of the rover's location. This information is then fed to the rover controller and action servers to be used for performing more advanced behaviors. To prevent inundating the main rover control with input and to enable more modular code construction, the main rover controller breaks off some of its responsibilities into "Action Servers." Each action server is responsible for performing a single operation and thus needs to directly listen to the relevant senor feeds when accomplishing its task.