IRoC-U

ISRO Robotics Challenge - URSC

Let's build a space robot

Proposal Report

by

LUNARATH
10963

January 2024

National Institute of Technology Warangal

Table of Contents

Introduction

Team Members	Expertise and Achievements
Pratyushraj Sharma (Project Lead)	• Team Lead in Successful Development of Surveillance Drone as part of Smart India Hackathon 2021. • Research and Development Skills in Hardware Systems, CAD simulations, Mechanical Design and Manufacturing, MATLAB simulations, Algorithm Development • Project Planning, Prototype and Product development.
Aditya Jana	• Patent in Mechanical Design of Surgical Skin Stapler • Proficient in Prototyping, Process Simulation
Rohan Karampuri	• Qualified for Stage 2 in the development of a space bot (Astro Tinker Bot) as part of the eYantra 2023-2024 program conducted by IIT BOMBAY. • Open CV, Path Planning Algorithms, UART protocol, Jetson Programming, Machine Learning Algorithms.
Bushra Shaik	• Team Lead in the development of a space bot (Astro Tinker Bot) as part of the eYantra 2023-2024 program conducted by IIT BOMBAY and Qualified for Stage2. • Implemented Sensor Integration in the development of a Waste Management Bot during the Hack Revolution event organised by ACES. • Embedded Systems, RISC V, RPi and Arduino programming, RF communication.
Sai SasiVardhan	• Participated in the eYantra Robotics Competition, contributing to the development of a rover capable of autonomous navigation in a simulated space environment. • Developed a 3 DOF Robotic Arm in the implementation of a Waste Management Bot during the Hack Revolution event organised by ACES • Open CV, Arduino, RPi and Jetson Programming
Saif	• Successful Development of Ambulance Drone as part of Firefly Hackathon • SolidWorks, Blender, Ansys
Afra Firdouse	• Arduino, RPi programming • PID controllers, Drone technology, Machine Learning Algorithms

Abdul Hafeez	• Participated in the eYantra contributing to the development of a rover capable of autonomous navigation in a simulated space environment. • ROS development and navigation stacks, Utilising Gazebo for robot simulation, OpenCV, Aruco marker detection.
Devi Sri	• Participated in the eYantra Robotics Competition, contributing to the development of a rover capable of autonomous navigation in a simulated space environment. • Experience with Aruco marker detection.
Uma Maheshwar	• Profound in Robotic Operating System and System Integration of multiple frameworks.

Our project LUNARATH, envisioned for IRoC-U 2024, is a perfect example to showcase the team's enthusiasm for robotics and responsibility to lead India's forefront in technological advancement.

We have come up with a vivid and detailed analysis of the problem statement and are ready to showcase a novel approach to it, which we believe will not only help us achieve victory in the competition but also help shape the aerospace industry around the world and help India achieve multiple milestones in the future.

Project Mangalyaan and Chandrayan were spectacular examples of what India is capable of in the field of aerospace. The splendid goals that these projects have achieved, along with the dedication showcased by the ISRO team in making them, are our very inspiration and motivation to take up this competition.

We have been moved by the enormous potential the field of space exploration holds. Our team definitely wants to put its technical capabilities into serving this country primarily and helping it reach new heights in whatever way possible. We believe IRoC 2024 will help us achieve this mission, and hence, we would want to seize this opportunity and prove our technical talents and patriotism.

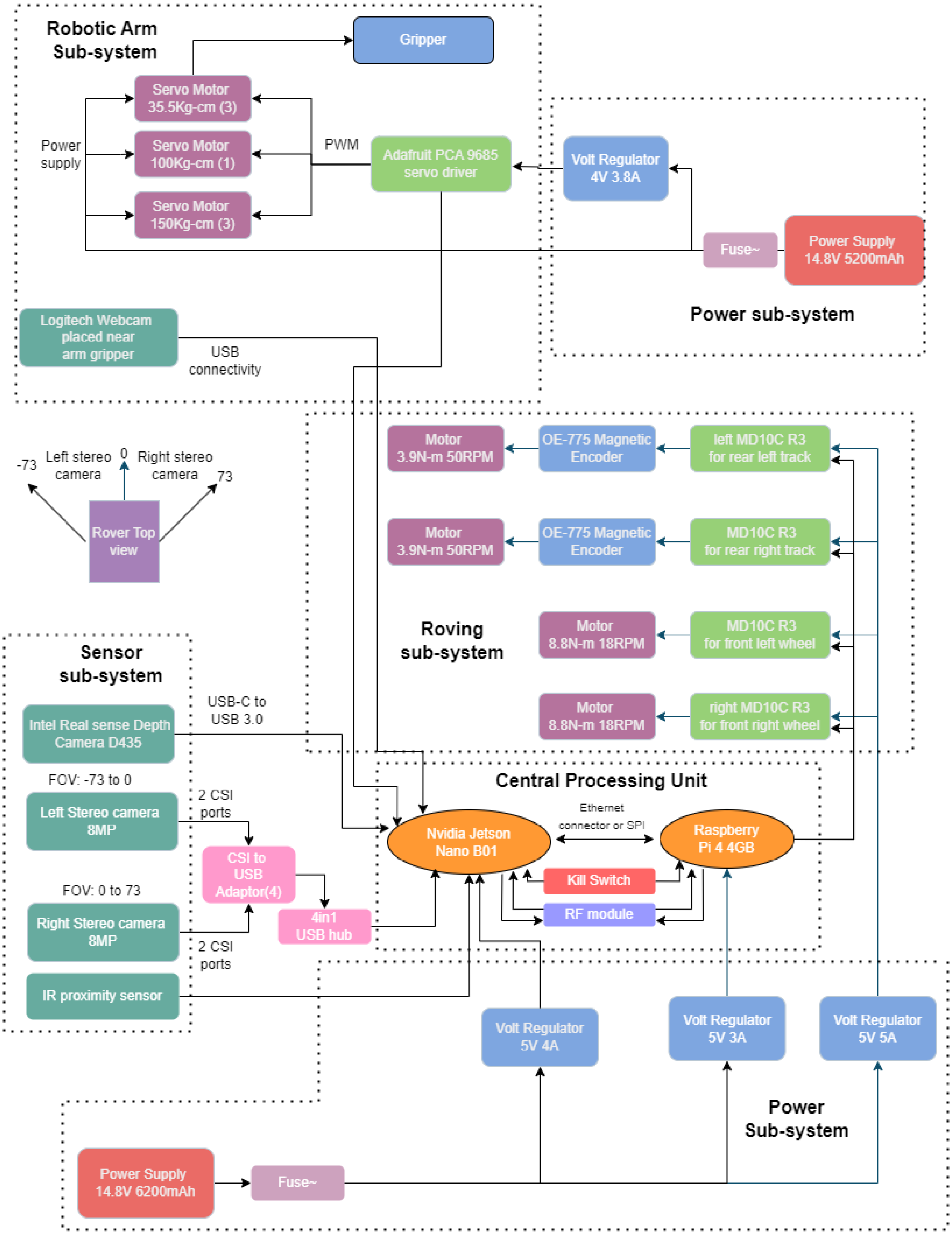
Task Requirements Analysis and Alignment of Technical Expertise:

We have gone through the IRoC-U 2024 project requirements and comprehensively understood the intricacies outlined. Our team carefully broke down each part of the task, making sure we got the whole picture.

Confident in our capabilities, we're geared up to deliver a stellar rover solution. With expertise in microcontroller programming, mechanical design, sensor integration, CAD modelling, power analysis, RF communication, ROS simulations, machine learning, computer vision, and prototyping.

Our dedication shines through in the project proposal, where we've methodically addressed every aspect, aligning our strategy with the specific needs of IRoC-U 2024.

System Architecture



The System Architecture consists of following subsystems:

System Architecture Subsystems:

Central Processing Unit:

- The Central Processing Unit (CPU) serves as the rover's main computational hub, interfacing with sensors through the USB Hub to collect and store input data.
- Responsible for executing algorithms, the CPU acts as the core processing unit for decision-making and control functions.

Sensor Sub-system:

- This subsystem integrates various sensors, including but not stereo cameras, Depth cameras and IR proximity sensor for comprehensive environmental perception.
- Sensor data acquisition is crucial for tasks such as mapping, obstacle detection, and localization during rover operations.

Robotic Arm Sub-system:

- The Robotic Arm Sub-system comprises actuators and joints controlled by the CPU, enabling precise and coordinated movements.
- Designed for manipulation tasks, the robotic arm executes commands for object retrieval, placement, and other dexterous operations.

Roving Subsystem:

- Responsible for the rover's mobility, the Roving Subsystem encompasses propulsion mechanisms, steering controls, and obstacle avoidance algorithms.
- This subsystem ensures the rover's ability to navigate varied terrains, responding to environmental changes (PID algorithm) and user commands.

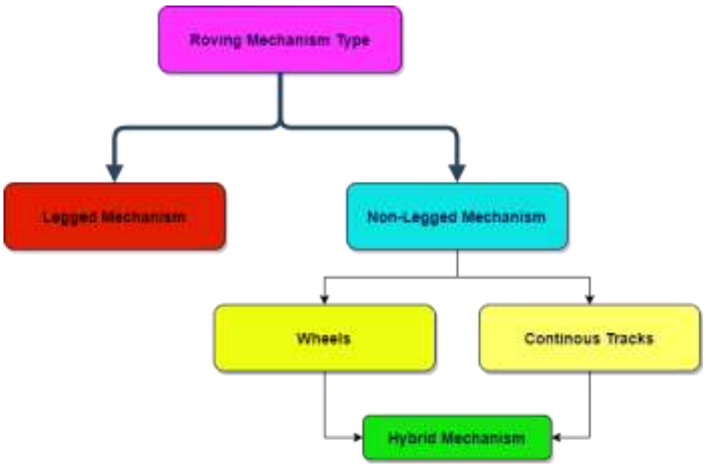
Power Subsystem:

- The Power Subsystem manages energy distribution, utilizing batteries or power sources to supply the required voltage and current using regulators.
- Ensures sustained operation of all subsystems, contributing to the rover's autonomy and efficiency during extended missions.

Roving Mechanism

The selection of roving mechanism depends on majorly on two broad category as internal factors and external factors. Internal factors are influenced by power consumption of the mechanism, number and geometry of contact points, centre of gravity, angle of contact, shape and size of the mechanism. Meanwhile, the external factors entirely depend on the working environment which includes structure of terrain, friction factor, quality of soil, mobility and other relevant factors.

The mechanism available for roving mechanism are legged, wheels and continuous tracks. The comparative study of the mechanism was discussed.



Legged Mechanism

- Walking robot is capable of crossing a crater so long as it reaches exceeds the width of hole.
- The main disadvantage is greater power consumption and lot of mechanical complexity. The design and the calibration of the legged mechanism is to complex for this project.
- This mechanism is **rejected to reduce complexity and to increase the speed** of the overall system throughout arena.

Wheels

The most common selection of wheel mechanism used by many existing rovers is a rocker-bogie arrangement.

Advantages:

- Better Stability when the contact points are high.
- Overall low Power Consumption
- Suitable for uneven and rough terrain
- Lesser Chassis Weight

Disadvantages:

- Less surface to area ratio, which reduces its mobility in loose or sandy terrain due to sinkage.
- Power Consumption increases drastically while turning as compare to the tracks, thus drawing power from batteries fluctuates more often.

Continuous/ Caterpillar Tracks

Advantages:

- High Surface to area ratio preventing it from sinking at a loose soil.
- Mobility at loose terrain increases due to lower sinkage and high surface area.
- The power consumption is almost uniform while turning from one direction to another.

Disadvantages:

- High Power Requirement, and large percentage of power is consumed to move tracks regardless of the terrain.

Considering all the parameters and to generate optimized solution a hybrid model is proposed for the given terrain.

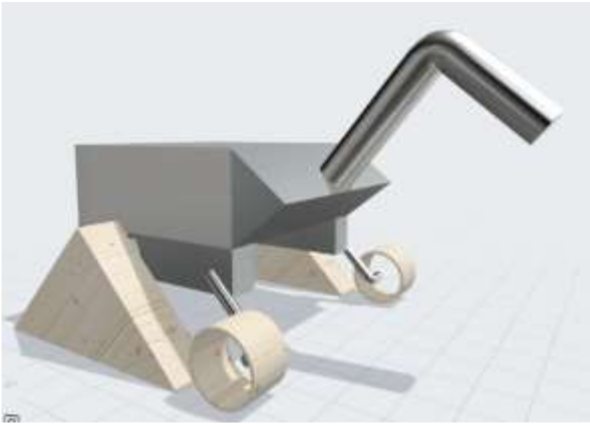
Hybrid Mechanism

The mechanism consists of having tracks at the rear wheel position and wheels at the front part, balancing the overall weight of the rover. The tracks will cover more than half of the body of rover and wheels will cover a small percentage of body weight.

- This design can operate in uneven terrain.
- The continuous plates at the offers enough traction to navigate in the loose soil.
- Hybrid mechanism will operate at intermediate power consumption between wheels and tracks mechanisms.
- Improves Navigation efficiency and suitable for required terrain.

The use of hybrid mechanism enables rover to navigate around the arena efficiently as compared to wheels and tracks design. It provides enough traction to hold its position on inclined surfaces. The track design has better mobility on the soft sand.

The rough diagram of the roving mechanism as follows:



Note: The provided model for the rover is just for getting an idea for the hybrid mechanism (It is not a CAD Model).

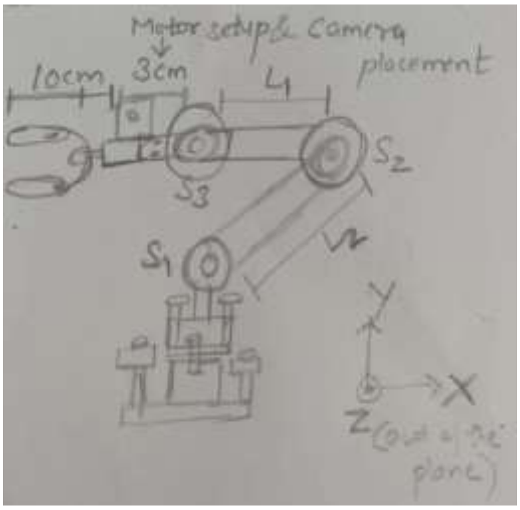
A triangular Shape Coninous tracks will be attached at the rear side of the rover, but for visualization a trianglur shape solid part is attached.

Mechanism for Sample Pick-and-Place Activity

Robotic Arm

For sample collecting a manipulator arm is designed with following specification.

Structure:



1. Degree of freedom: 4
    a. The minimum required DOF which can perform the task will be 3 but the size of the robotic arm will be big enough which add extra weight and volume to the rover.
    b. To avoid this an extra DOF is added to make a compact and light weight arm which can be easily handle by the rover and occupies less volume for storage.

2. Torque Calculation of each joint provides us with the torque requirement for servo motors. The required ratings are as follows:
    a. Base joint rotation servo = 150kg.cm (rotation in X-Z plane)
    b. S1 = (250-300) kg.cm (rotation in X-Y Plane)
    c. S2 = (80-100) kg.cm (rotation in X-Y plane)
    d. S3 = 35 kg.cm (rotation in Y-Z plane)
    e. Gripper servo = 35kg.cm

    Note: The above plane of rotation of each servo is referenced to the above image assuming other servo motors rotation as constant.

3. Material of construction will be a 3D printed (PLA)lightweight material with higher strength.

4. The Gripper structure is based on **Gear-and-Rack Actuation** which will be functional by the movement of attached servo motor which provides an optimum working field of view (FOV).
    a. Other mechanisms like linkage actuation will become complex and need more precise operation, so to avoid complexity we discarded this mechanism.
    b. Non-mechanical grippers like;
        i. Vacuum Cups won't be efficient as the operation terrain will be a sandy terrain and field will be covered with dust which reduces the efficiency of suction cups.
        ii. Magnetic grippers are avoided as sample is not specified that it can be influenced by magnetic field.
        iii. Any other type of gripping mechanism is irrelevant to us.

Robotic Arm Working Visualization:

After the rover reaches near the red colour cylindrical block the arm gets the signal with the help of the camera which is above the gripper; the camera calculates the coordinate of the red colour cylindrical block and sends it to the inverse kinematic function which calculates the pitch, angles of the servo motor for each joint and the centroid of block. Then the arm catches the red-coloured cylindrical block with the help of the gripper and then place it on the rover.

Emergency Response System

Possible Emergency Situations	Identification	Response System
Collision with unwanted Objects: Ex - Big Cubes	• Collision Monitor Node in nav2 • Time of Flight Sensors (Ultrasonic Sensor preferably)	• Emergency Braking System
Wrong sample placement	• Camera attached to the gripper	• Using open CV and re running the algorithm (inverse kinematics)
MCU malfunctioning	• Watchdog Timer	• Watchdog Timer
Communication Disruption	• CRC (Cyclic Redundancy Check) • Signal Strength Indicator	• Built-in Auto re-transmission • communication watchdog
Mechanical malfunctioning	• Positional Encoders	• Custom Recovery Behaviour using nav 2 using the data available from the encoder sensors. • Emergency Halt
If the rover tries to get out of Arena	• Visual Identification	• Remote control Override

1. **Emergency Braking:**
   - **a.** A mechanism that triggers an immediate halt to the rover's movement in emergency situations.
   - **b.** For preventing collisions or further unintended movements that may pose a threat to the rover or its surroundings.
2. **Watchdog Timer:**
   - a. Although the Raspberry Pi and Jetson Nano have an in-built watchdog timer, this has a poor reputation.

RTC WatchDog HAT for Raspberry Pi/Jetson Nano:

- Standard Pi 40 PIN GPIO extension header, fits Raspberry Pi series boards and Jetson Nano.
- Onboard TSL25911FN digital ambient light sensor, for measuring IR and visible light; Onboard BME280 sensor, for measuring temperature, humidity, and air pressure
- Onboard SGP40 sensor, for detecting ambient VOC; I2C bus, allows retrieving data by just using two wires

3. **Auto re-transmission for communication Disruptions**
    a. Enabled an automatic retransmission mechanism for communication disruptions.
    b. When data transmission failures occur, the system automatically attempts to resend the data, enhancing the reliability of communication links and ensuring that critical information reaches its destination.
4. **Remote Control Override:**
    a. Utilising RF communication ensures seamless remote-control override in our rover system, enabling operators to intervene and navigate the rover manually from mission control of Dynamic Mission Scenarios.

Kill Switch Implementation

- A physical Switch connected to the GPIO pins on both the MCUs through pull down resistors (10K ohm and 20K ohm).
- When the kill switch is pressed, the GPIO pins on both the Raspberry Pi and Jetson Nano detect a rising edge.
- The pull-down resistors help ensure that the GPIO pins have a default low state.
- The rising edge triggers the respective shutdown functions in software, initiating a graceful shutdown for each MCU.

Hardware identification

No.	Hardware Details	Subsystem	Category	Quantity Needed	Justification/purpose of Chosen Type
1	NVIDIA Jetson Nano B01	CPU	MCU	1	To implement vision-based algorithms
2	Raspberry Pi 4 Model B	Roving Sub-system	MCU	1	To control rover using different algorithms and Sensors
3	Cytron MD10C R3 Motor Driver	Roving Sub-system	Motor Driver	4	Motor control and empowers motion
4	Adafruit PCA9685 16 channel PWMservo driver	Robotic Arm Sub-system	Servo driver	1	Servo control and empowers motion
5	IMX291-83 Stereo Camera	Sensor Subsystem	Sensor	2	Depth precision, Double image understanding
6	Intel Realsense Depth camera D435	Sensor Sub-system	Sensor	1	Precise depth and RGB capture for detailed arena understanding to achieve SLAM
7	Logitech C270 Digital HD Webcam	Sensor Sub-system	Sensor	1	Mounted on robotic arm to pick objects using OpenCV
8	Orange OT6560 (60kg.cm/120 degree)	Robotic Arm Sub-system	Servo Motor	1	Precise movement, lightweight, energy-efficient
9	DS51150 Servo Motor High Torque (150kg.cm)	Robotic Arm Sub-system	Servo Motor	3	Precise movement, Lightweight, Energy efficient
10	Orange OT5330M (35.5kg.cm/ 180 degree)	Robotic Arm Sub-system	Servo Motor	3	Precise movement, lightweight, energy-efficient
11	Orange Planetary GearDCMotor 12V 50 RPM 392.4 N-cm	Roving Sub-system	Geared DC Motor	2	Increased torque, efficiency, speed, power boost
12	Orange Planetary Gear DC Motor 12V 18 RPM 882.9 N-cm	Roving Sub-system	Geared DC Motor	2	Increased torque, efficiency, speed, power boost

13	OE-775 Hall Effect Two Channel Magnetic Encoder	Roving Sub-system	Encoders	2	Feedback, and control for accurate motion in systems
14	Switching Voltage Regulator 14.5V to 5V 4A 14.5V to 5V 3A 14.4V to 5V 5A 14.5V to 4V 4A	Power Subsystem	Power	5	Precise voltage control, reduced power dissipation
15	SD card 64GB	CPU	MCU	1	Ample storage, efficient data handling, accommodates complex AI models
16	Micro USB to type A cable from Jetson to computer	CPU	MCU	1	Convenient data transfer,
17	USB to CSI Adaptor for cameras	Communication System	Camera	4	Enable USB to cameras to connect seamlessly to CSI-equipped device
18	USB Hub 4 in 1 3.0 USB	Communication System	Camera	1	Multiple connection of camera to CPU
19	Orange 14.8V 5200mAh 40C 4S Lithium Polymer Battery Pack	Power Subsystem	Battery	1	High and sufficient capacity and voltage
20	Orange 14.8V 6200mAh 35C 4S Lithium Polymer Battery Pack for Rover	Power Subsystem	Battery	1	High and sufficient capacity and voltage
21	Kill Switch	Power System	Switch	1	
22	Fuse	Power System	Circuit Breaker	4	
23	Wireless NIC for Wifi or Bluetooth	Communication		1	

24	Radiolink AT10 II 2.4GHz 12CH RC Drone Transmitter	Communication		1	Emergency control of rover and Arm
25	Radiolink R12DS 2.4GHz RC Receiver 12 Channels SBUS/PWM	Communication		1	Emergency control of rover and Arm
26	iMax B6 Digital LiPo	Electrical Power Subsystem	Charger	1	digital display and can charge multiple batteries simultaneously

- Our rover is equipped with advanced technologies, including depth cameras, stereo cameras and sensor integration using extended Kalman filter algorithm and some ROS2 software for roving mechanisms.
- These processes are handled by a Jetson, making decisions about the rover's movement. Information is then sent to a Raspberry Pi, which, in turn, communicates with four motor drivers to control the rover's motion.
- The rover is smart enough to avoid 30 cm cubes and 20cm obstacles while safely traversing over 15 cm cubes and 10cm craters. Upon reaching a specific waypoint (WP) region, the rover, guided by stereo cameras, stops 20cm away from a red cylinder.
- It instructs a webcam on the gripper to search for the red cylinder using OpenCV. If found, a robotic arm lifts the cylinder and places it on the rover before resuming movement.
- Additionally, the rover utilises PID control for efficient climbing on inclines. It searches for a blue cylinder using stereo cameras, stops 20 cm away when found, and places a red sample into the blue cylinder using a gripper-mounted webcam and OpenCV.

Software identification

Open-Source Framework: ROS2 is a middleware framework designed for robot development, OpenCV is a computer vision and machine learning Library, Gazebo is a 3D robotics simulator, Nav2 and SLAM.

Algorithm are discussed as follows:

S No.	Software & Algorithm	Description (Justification)	Available hardware
1	ROS2 Iron Irwini	It is an open-source software which includes all the tools and libraries required for the development of the rover mechanism.	Raspberry Pi 4 Model B
2	Nav 2 package behaviour trees algorithm	Nav2 uses behaviour trees to create customised and intelligent navigation behaviour via orchestrating many independent modular servers.	Raspberry Pi 4 Model B
2	SLAM an MCL algorithm	Localization and Mapping of the arena	Intel Real sense depth camera & Stereo Cameras
3	Open CV integration with an ML model	To detect the colour of waypoint, Final Point, cylinder as well as the detection of cubes & craters in the arena and act accordingly.	Intel Real sense depth camera RGB-D 435, Logitech web camera
5.	PID Algorithm	for precise controller of rover, stability, maintain specific position	Encoders and motors
6.	Extended Kalman Filter	To integrate all the sensors used in the rover.	CPU
7.	Visual Odometry	Determine how the rover has moved over time by tracking features in consecutive images	All cameras
8.	Vector Field Histogram (VFH)	computing the obstacle-free steering directions for a robot based on camera readings	All Camera
9.	Simple api commander	simple API commander is used for navigating our rover confidently in any environment by using python	Raspberry Pi 4 Model B
10.	Arm Coordination	Integrate algorithms to coordinate the rover's navigation with the robotic arm's	Adafruit PCA9685

		actions, ensuring smooth and safe object manipulation.	

ROS 2 "IronIrwini" may offer advantages like improved real-time capabilities, enhanced security features, and middleware abstraction. Its distributed system design, community support, and flexible platform compatibility make it suitable for advanced autonomous rover projects compared to earlier ROS versions.

Nav2 behaviour trees have emerged as a powerful tool for crafting complex and adaptive behaviours. These trees offer a structured way to orchestrate a diverse set of actions, enabling robots to navigate effectively in dynamic and challenging environments.

Integrating **OpenCV** with machine learning models for cubes and craters detection, using two stereo cameras and an RGB-D camera on the rover, enhances obstacle identification accuracy and depth perception. This multi-sensor approach enables robust recognition in diverse environments, crucial for precise autonomous navigation and avoidance in complex terrains, ensuring the rover's safety and efficiency.

Using a **PID controller** for wheel movement with wheel encoders ensures precise and adaptive control, compensating for errors and minimising settling time. The PID algorithm's versatility, stability, and real-time response contribute to achieving smooth and accurate rover movements in dynamic environments.

Extended Kalman Filter: The extended Kalman filter (EKF) is an approximate filter for nonlinear systems, based on first-order linearization of the process and measurement functions. This is the non-linear variant of the Kalman filter. Compared to other algorithms, the EKF has the potential to be the most flexible and accurate.

Visual odometry tracks features, matches positions, and estimates motion, iterating over time and refining trajectory using techniques like bundle adjustment. The evolving trajectory is crucial for tasks like simultaneous localization and mapping (SLAM) in autonomous navigation.

The vector field histogram (VFH) algorithm is used in computing the obstacle-free steering directions for a robot based on camera readings. It involves several steps including processing the depth data, obstacle detection, generation of polar histogram. Depth data can be used to compute polar density histograms to identify obstacle location. VFH will also help rover to get local path plan and follow a target direction while avoiding obstacles and craters.

The arm works on the principle of **Inverse Kinematics** and detects the object using **OpenCV**.

At Destination: Forward Kinematics is applied where the angles of the servo motors are decided to find the coordinate to place the red coloured cylindrical block in blue coloured platform.

- **Camera Initialization:**
    - The program starts by initializing the camera connected to the Jetson Nano. The camera captures images in real-time.
- **Computer Vision Setup:**

  - Computer vision is employed to process the captured images. The program converts the images to the HSV colour space, a colour representation that is particularly useful for colour-based segmentation.
- **Red Object Detection:**
  - The program defines a range of HSV values corresponding to the color red. By thresholding the image based on these values, it identifies regions that likely contain red objects.
- **Contour Analysis:**
  - The program then identifies contours in the threshold image. Contours represent the boundaries of distinct shapes in the image.
- **Centroid Calculation:**
  - From the contours, the program identifies the largest contour, likely corresponding to the main red object in the scene. The centroid of this contour (the centre of mass) is calculated to determine the position of the red object in the image.
- **Inverse Kinematics:**
  - The centroid coordinates, representing the position of the red object in the image, are then passed to the inverse kinematics calculation. This involves solving mathematical equations to determine the joint angles necessary for the robotic arm to place its end-effector at the identified 3D coordinates.
- **Robotic Arm Movement:**
  - The calculated joint angles are used to control the robotic arm. The arm adjusts its configuration based on the inverse kinematics solution, attempting to align its end-effector with the detected red object.
- **Visual Feedback:**
  - Throughout this process, the original camera frame and a mask highlighting the red object are displayed for visual feedback. This allows monitoring and verification of the robot's interaction with the environment.

Hardware and Software Realization Plan

Sensors and MCU Realisation:

No.	Hardware Details	Procurement Source (Market)	Specifications/ Realisation Plan	Quantity	Estimated Cost per component
1	Nvidia Jetson Nano B01	Market	Quad-Core Arm® Cortex®-A57 MPCore CPU at 1.43Hz 128-core NVIDIA Maxwell™ architecture GPU` 4GB 64-bit LPDDR4 25.6GB/s up to 4K video encode and decode	1	30,000

			5V 4A power supply required		
2	Raspberry pi 4	Market	64-bit Quad Core Cortex A72 Processor 4 Gb Lpddr4 Ram 2 Micro HDMI Ports, 2 USB 3.0 Ports ,2 USB 2.0 Ports, Giga-bit Ethernet Port 5V,3Amp USB Type C power Supply Required	1	5,600
3	Cytron MD10C R3 Motor Driver	Market	Bi-directional control for 1 brushed DC motor, single channel Support motor voltage ranges from 5V to 30V. Maximum current up to 13A continuous 3.3V and 5V logic level input.	4	1,150
4	Orange Planetary Geared DC Motor for tracks	Market	Operating Volt: 12V DC Torque: 392.4 N-cm Rated Speed: 50 RPM Rated Current: 4.28 A Rated Power: 50.97 W Gear Ratio: 99.5: 1	2	2,355
5	Orange Planetary GearDC Motor for wheels	Market	Operating Volt: 12V DC Torque: 882.9 N-cm Rated Speed: 18 RPM Rated Current: 4.8 A Rated Power: 58.76 W Gear Ratio: 264: 1	2	2,355
6	OE-775 Hall Effect Two Channel Magnetic Encoder	Market	Supply Voltage: 4-24V Supply Current: 14mA	2	370

7	Intel Realsense Depth camera D435	Market	Ideal range: 0.3m to 3m depth field of view: 87° × 58° (±3°) depth accuracy: <2% at 2m RGB sensor resolution: 2MP	2	38,000
8	IMX219-83 Stereo Camera	Market	angle of View: 83/73/50 degree (diagonal/ horizontal/vertical) Megapixels: 8MP	1	5,350
9	ORANGE 6200/4S-35C	Market	Capacity: 6200 Mah Constant Discharge: 35C. Output voltage:14.8V	1	6,500
10	SanDisk Ultra SDcard	Market	Memory Storage Capacity: 64GB Up to 140MB/s transfer speed	1	500
11	Type A to Micro USB Braided Cable	Market	480 Mbps Data Transfer Speed Rated value: 3A/18W	1	300
12	Jetson Power supply	Market	20W 5V 4A power supply	1	650
13	Ethernet Cable	Market	Cat 7 High Speed	1	560
14	step up-down Voltage Regulator	Market	input volt DC5.5-30V, output volt DC 0.5-30V, can be arbitrary adjustable; output current range is 0-3A, can be arbitrary adjustable	5	1000
15	Fuse	Market	easily replaceable	4	100
16	Wireless NIC for Wifi or Bluetooth	Market	Bands: 2.4GHz / 5GHz Speed: 300Mbps / 867Mbps WiFi protocol: 802.11ac Bluetooth version: 4.2	1	1,700
17	iMax B6 Digital LiPo Battery Charger	Market	Input voltage: 11-18V Charge current range:0.1- 5A discharge current range:0.1-1A	1	2,000
18	Radiolink AT10 transmitter	Market	Operating Voltage (VDC):7.4-12 V No of channels:12 transmitting frequency :2.4GHz	1	11,130

19	Radiolink R12DS RC Receiver	Market	Operating Voltage: 3~10V Operating Current: 38-45mA@5V receiving frequency: 2.4GHz	1	2,000
20	Kill Switch	Market	Push mechanism	1	200

Sum: Rs 1,29,830

Robotic arm

no.	Hardware Details	Procurement Source (Market/Fabrication/3D Printing)	Specifications/ Realisation Plan	Quantity	Estimated Cost
1	Logitech C270 Digital HD Webcam	Market	Crisp HD 720p/30 fps 55° field of view	1	2000
2	Orange OT6560	Market	Input Voltage Range (VDC): 4V-8.4V Rated Torque: 80Kg.cm @ 7.4V Maximum Operating Current (A): 3.8A Rotational degree: 180º	1	7500
3	DS51150 Servo Motor High Torque 150Kg	Market	Input Voltage Range (VDC): 10-12.6V Rated Torque: 150-165 Kg-cm Maximum Operating Current (A): Rotational degree:180	3	9,000
4	Orange OT5330M	Market	Input Voltage Range (VDC): 4V-8.4V Rated Torque: 35.5 Kg.cm @ 7.4V Maximum Operating Current (A): 3.8A Rotational degree: 180º	3	1800

5	Adafruit I2C interface – PCA9685	Market	16-Channel 12-bit PWM Operating voltage: 2.3V to 5.5V	1	450
6.	Orange Lithium Polymer Battery Pack	Market	Supply Voltage: 14.8V Capacity: 5200mAh C-number: 40C No. of cells: 4S	1	5200

Sum: Rs 47,350

Body Design:

The rover chassis and tracks will be constructed using aluminium rods and sheets, while the robotic arm will be 3D printed using PLA filaments, which cost approximately 2500. Given the availability of 3D printers in our college, the cost of the body design cannot be specified. The remaining expenses for printing and welding are estimated to be under 30000. The choice of materials, including aluminium for robustness and PLA for 3D printing versatility, aligns with cost-effective construction. The in-house 3D printing capabilities contribute to minimizing expenses and enhancing flexibility in the fabrication process, ensuring efficient resource utilization for the rover project. (This is an assumed cost for building the complete body)

The miscellaneous cost required to support the building of proposed rover which is important to the project is assumed to be Rs:10,000 (Tools, equipment and accessories or development of prototype)

Total Budget of the Rover: 1,77,180 +30000 + 10,000= 2,17,180
(Approximately 2.1 Lakhs)

NOTE: The above price details of the component and system are prepared by assuming the current market price. If any fluctuation in price of any component changes, then the total budget of the rover also deviates from the proposed mechanism.

Software Implementation Strategies

1. Navigation and SLAM:

- Utilize the Nav2 stack with Adaptive Monte Carlo Localization (AMCL) for robust mapping and localization.
- Fuse data from stereo cameras, depth cameras, and Time of Flight sensors using Nav2's sensor fusion framework for accurate obstacle detection and crater avoidance.
- Implement path planning algorithms that consider your rover's arm reach and manipulability constraints to ensure smooth object pickup and placement.

- Integrate a collision checker within the navigation stack to prevent collisions with obstacles during autonomous movement.

2. Arm Manipulation and Object Detection:

- Control your arm through inverse kinematics calculations for high-level movement commands and trajectory.
- Develop an OpenCV-based object detection algorithm to identify the cylindrical object using its shape, colour, and texture.
- Consider using background subtraction or template matching techniques for robust detection in varying lunar lighting conditions.
- Estimate the precise pose (position and orientation) of the object using stereo camera and depth sensor data for accurate arm positioning.

3. Additional Considerations:

- Implement error handling mechanisms for sensor failures, unexpected obstacles, and manipulator errors.
- Optimize software for efficiency to minimize resource consumption (battery, processing power) during the competition.
- Thoroughly test your software in simulated environments using Gazebo or other robotics simulators to validate navigation, arm control, and object detection functionalities.
- Develop contingency plans for potential scenarios and ensure your rover can adapt to unforeseen circumstances.

4. Software Architecture:

- Modularize your code by separating navigation, arm control, object detection, and other functionalities into independent modules for easy maintenance and code reuse.
- Utilize existing libraries and tools for common robotics tasks like sensor processing, perception, and control whenever possible to save development time and effort.
- Implement a clear communication interface between modules to ensure seamless data exchange and coordination.

5. Testing and Validation:

- Perform unit testing of individual modules to ensure proper functionality.
- Conduct integration testing to verify how different modules interact with each other.
- Extensively test your software in Gazebo simulations with various Moon-like terrain features and scenarios.
- Refine and optimize your algorithms based on the results of your testing and simulations.

Test Plan

Robotic Arm:

- **Load Bearing Capacity:**

Test: Determine the maximum payload the arm can safely lift and manipulate.

1. Gradually increase the weight attached to the end-effector until performance degrades or failure occurs.

Pass/fail criteria: Maximum payload should meet design requirements without compromising stability or accuracy.

- **Servo Motors and Specifications:**

Test: Verify the performance of servo motors and their compatibility with the control system.

1. Test individual motors for speed, torque, accuracy, and response time.
2. Assess communication and control accuracy between motors and control boards.
3. Pass/fail criteria: Motors should meet datasheet specifications and operate smoothly with the control system.

- **Inverse Kinematics:**

Test: Evaluate the accuracy and efficiency of inverse kinematics algorithms.

1. Program the arm to reach various target positions in 3D space.
2. Measure the time required to calculate joint angles and the precision of final positioning.

Pass/fail criteria: Inverse kinematics should produce accurate joint angles within acceptable time limits.

Additional Tests:

- Power consumption: Measure the arm's power draw under different load and movement conditions to ensure it aligns with rover power constraints.
- Endurance testing: Conduct repetitive movements over extended periods to evaluate long-term wear and tear on components.
- Integration testing: Verify seamless interaction and communication between the arm and other rover systems (e.g., power, control, navigation). RF channel
- Sample creation in real world and performing the same task as performed in simulation.

Navigation System of Rover:

- **Path planning:**

Test objectives:

To verify path generation, global and local path planning capabilities of rover.

Test procedure:

1. Design and plan a path for rover and command to navigate autonomously.

2. Monitor the robot's real-time path and check for deviations in rover navigation while moving in desired path.
3. Check if any inaccuracies while navigating through obstacles, craters, terrain.

Pass/Fail criteria: The rover should be able to follow the path with minimal deviation.

- **Localisation**:

Test objectives:

To check the accuracy of MCL algorithm to various environments, robustness to sensor noise and capabilities

Test procedure:

1. Performing simulation with considering local arena, visualising in Rviz that how rover can be able to localise itself in arena with depth data available from
main camera (RGB-D) and wheel encoders.

2. Taking stereo cameras, RGB-D cameras data and performing sensor fusion.
3. Comparing the accuracy of localisation with respect to the local arena in both the cases.
4. Checking if there are any irregularities in multiple sensor readings and applying smoothing algorithms.

Pass/Fail criteria:

The rover should be able to estimate it's pose accurately in local arena with both cameras and wheel encoders with minimal delays and errors.

- **Mapping test plan**

Test objective: Evaluate the mapping process's effectiveness in various scenarios (such as speed and environment complexity).
Test procedure:

1. Keep track of the robot's progress as it creates a map of various sized and sophisticated areas, such as an obstacle course, an empty room, and outdoor terrain.
2. While mapping, keep track of how much memory, computing power, and batteries are being used. Examine mapping times and resource usage in various scenarios.
3. Compare various URDF files for accuracy while looking into its various scenarios.

- **Motion control test plans**

BASIC MOVEMENT
Test objective: verify the rover's basic movement (forward, backward and turning) accurately and smoothly
Test procedure:

1. Commanding our robot to move specific points and performing various turnings.
2. Measuring the travel distance using odometry and cross checking

3. The movement errors must fall within permissible tolerances (e.g., angle error 5°, distance error 1%) in order to pass or fail.

SPEED CONTROL
Test objective: Evaluating the rover's ability to maintain desired speed during straight and turning movements
Test procedure

1. Give the robot varying speeds to travel on flat areas and slope surfaces.
2. Measure the actual speed using odometry or external sensors. Compare the measured speed with the required speed to assess accuracy.
3. To evaluate response, repeat the test with abrupt changes in speed orders.
4. Pass/Fail Criteria: Deviations in speed must stay within acceptable limits, which might be ±10% of the desired speed.

OBSTACLE AVOIDANCE
Test objective: Test for robots' ability to detect and avoid obstacle using sensors and checking the mapping data
Test procedure

1. Place several kinds of sizes and shape obstacles in the robot's path. Observe how the robot behaves when it gets closer to the obstacles.
2. Check to see if the robot has avoided the obstacles successfully by employing the proper movements (e.g., halting, turning).
3. Pass/Fail Criteria: The robot must continually avoid running into objects and exhibit cautious, considerate behaviour.

- **Integration with other functionalities:**

1. Test Objective: Ensure seamless integration of the map with other robot functionalities (e.g., navigation, sensor fusion, obstacle avoidance).

Test Procedure:

2. Command the robot to navigate to specific locations within the map and observe its path planning and execution.
3. Simulate obstacle encounters and verify the robot's ability to use the map for obstacle avoidance.
4. Test other functionalities that rely on the map (e.g., target tracking, area exploration) and ensure proper integration with simulating different kind of obstacles at the same time.

Pass/Fail Criteria: All robot functionalities should operate seamlessly with the generated map, without errors or conflicts

- **Images to Enhance Content Verification Tests:**

  - RealSense Depth Camera D435:

    Verify depth sensing capabilities in varying lighting conditions.
    Evaluate the camera's ability to capture and transmit real-time depth data.
    Validate the integration with the rover's overall sensor system for coherence.

- Nav 2 Package Architecture:

  Review the Nav 2 package diagram for clarity and completeness.
  Simulate software components' interactions in a controlled environment.
  Perform stress testing to evaluate the package's resilience under varying loads.

- Navigation Test Environment:

  Confirm rover's ability to navigate around obstacles using sensor data.
  Evaluate marker detection and interpretation by the rover's software.
  Assess the rover's responsiveness and adaptability to changes in the test environment.

System Specifications

S No.	Name	Description
1.	Weight	Approx.30Kg
2.	Length, breadth and height	70cm X 70cm X 55cm
3.	Arm extended length	70cm
4.	Arm Payload	500-600gm approx.
1.	Arm Weight	Approx. 12-15kg
6.	3D printing Material	PLA
7.	Power Supply	Rechargeable LiPo Batteries
8.	Radio frequency Joystick module	Manual Rover Control
9.	Safety Measures	Kill switch, Fuses, Regulators and other Emergency response algorithms

Project management

No.	Task	Main Responsibility	Completion of Task	Secondary Responsibility
1	Navigation and obstacle detection	Team Member 3 (Umamaheshwar)	20th Mar	Team Member 7 (Abdul Hafeez)
2	Robotic Arm Pick &Drop mechanism(software)	Team Member 2 (SasiVardhan)	20th Mar	Team Member 9 (Devi Sri)
3	Hardware Procurement	Team Member 5 (Bushra)	-	Team Member 8 (Afra)
4	Cad model of robotic arm and Assembly	Team Member 6 (Saiful)	25th Jan	Team Member 4 (Aditya)
5	CAD model of complete Rover (except robotic arm)	Team Member 4 (Aditya)	25th Jan	Team Member 6 (Saiful)
6	Arena Planning & Prototyping	Team Member 1 (Rohan)	1st May	Team Member 2 (SasiVardhan)

Novelty in the overall proposal

Roving Mechanism:

- The mechanism we selected considering the terrain and working environment the proposed mechanism of **hybrid model** is developed by the innovative ideas of our team members. We read the relevant research papers and find out the optimum model on paper and we are confident that the proposed mechanism will work efficiently throughout the arena.

Body Design

- The body designs are made is such a way that robotic arm will going to merge itself on the rover body making it a box shape rover body. At the time of operation, it will emerge through the body and after completion it get back to its starting position.

The Further Upgradation to the body is done as per the requirement and fixing the possible errors faced while manufacturing it.

THANK YOU