

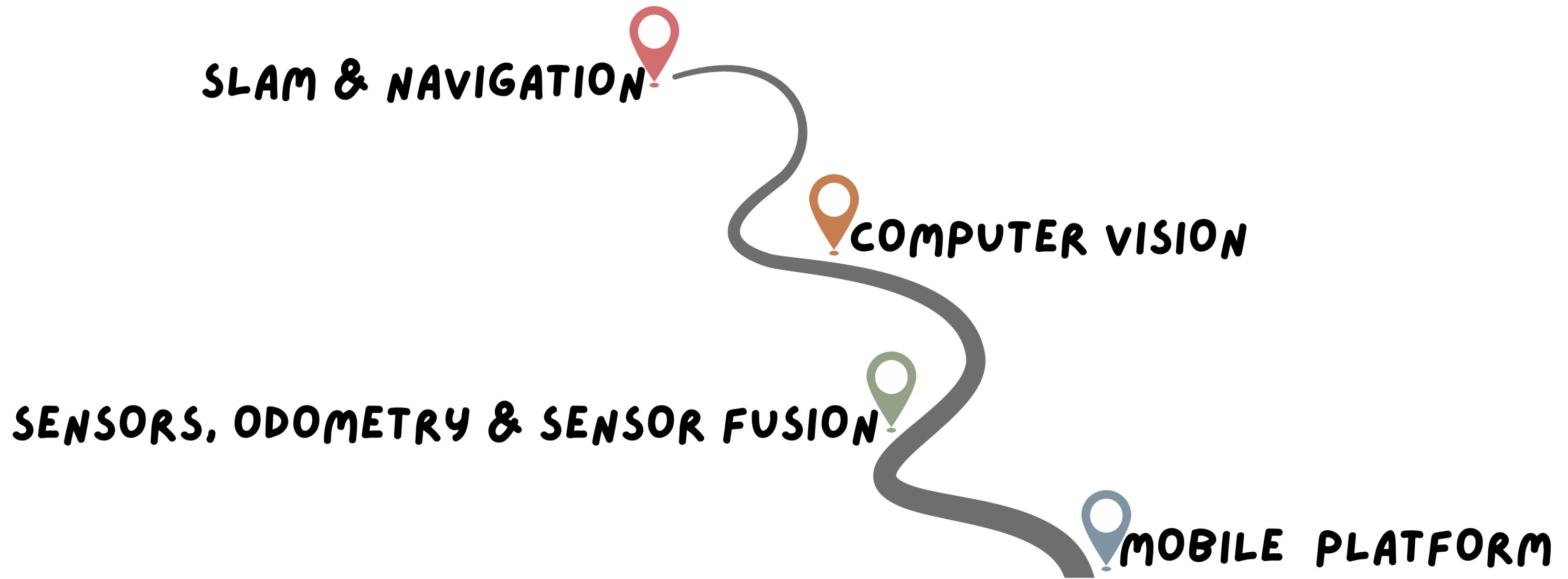


Knights Technical Presentation

Robotics Dojo 2025



Amos Oniare
Dismas Karimi
Gareth Kipkoech
Brian Fundi

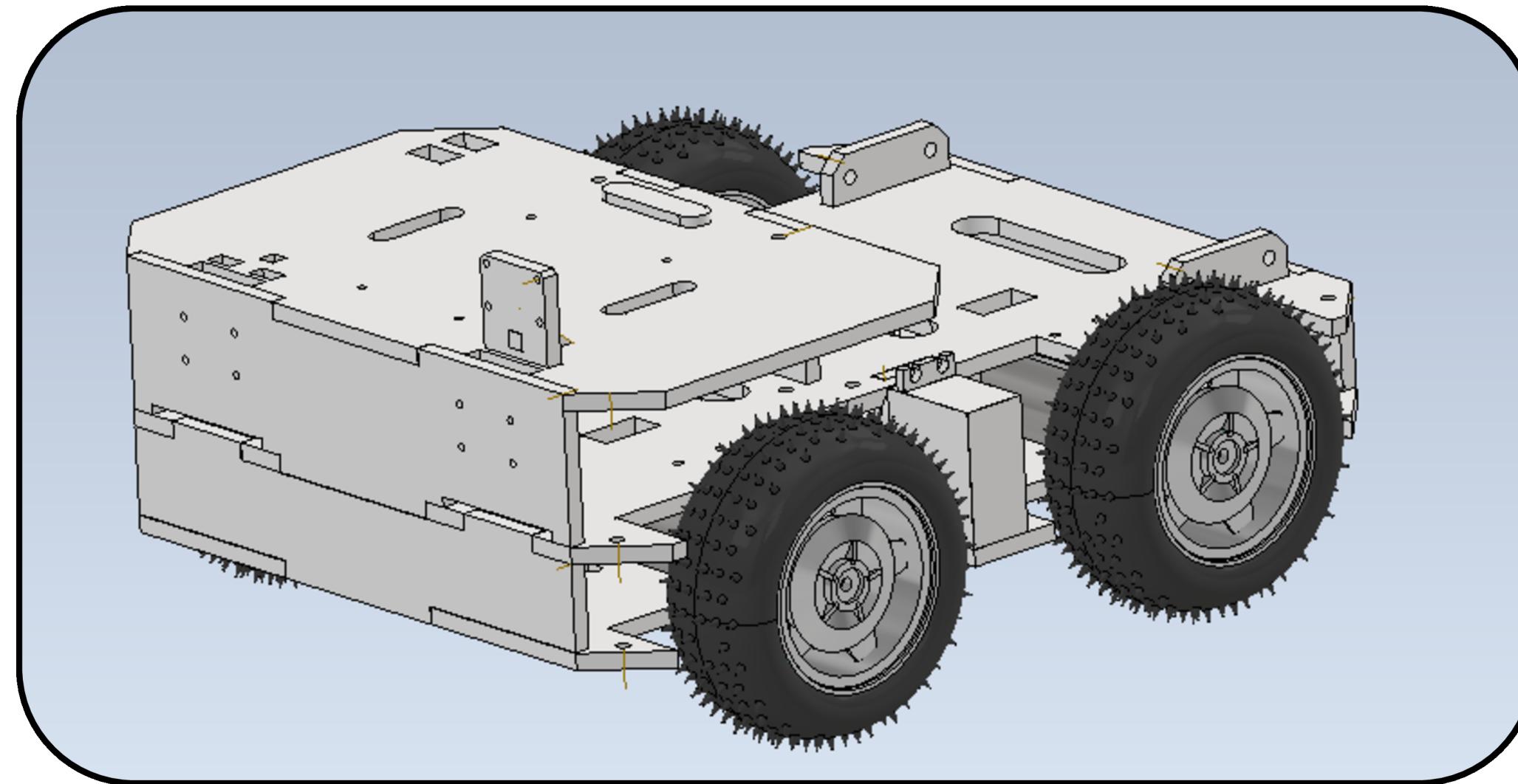


MOBILE PLATFORM: MECHANICAL

Drivetrain: 4-Wheel Differential Drive

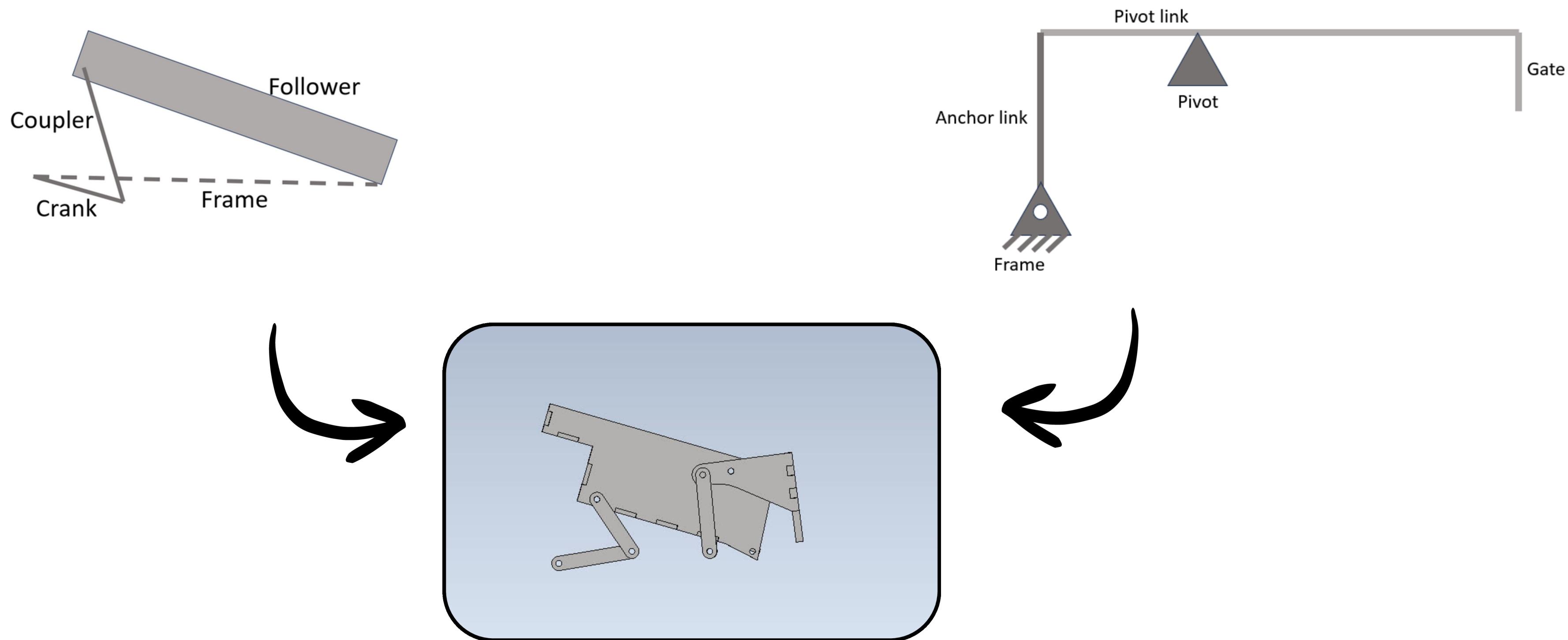
Frame Material: PMMA

Aimed to minimize track, wheelbase, height



MOBILE PLATFORM: MECHANICAL

Offloading: Single-servo actuation



MOBILE PLATFORM: ELECTRONICS

Sensors: 4 rotary encoders, IMU, Lidar, Pi Camera

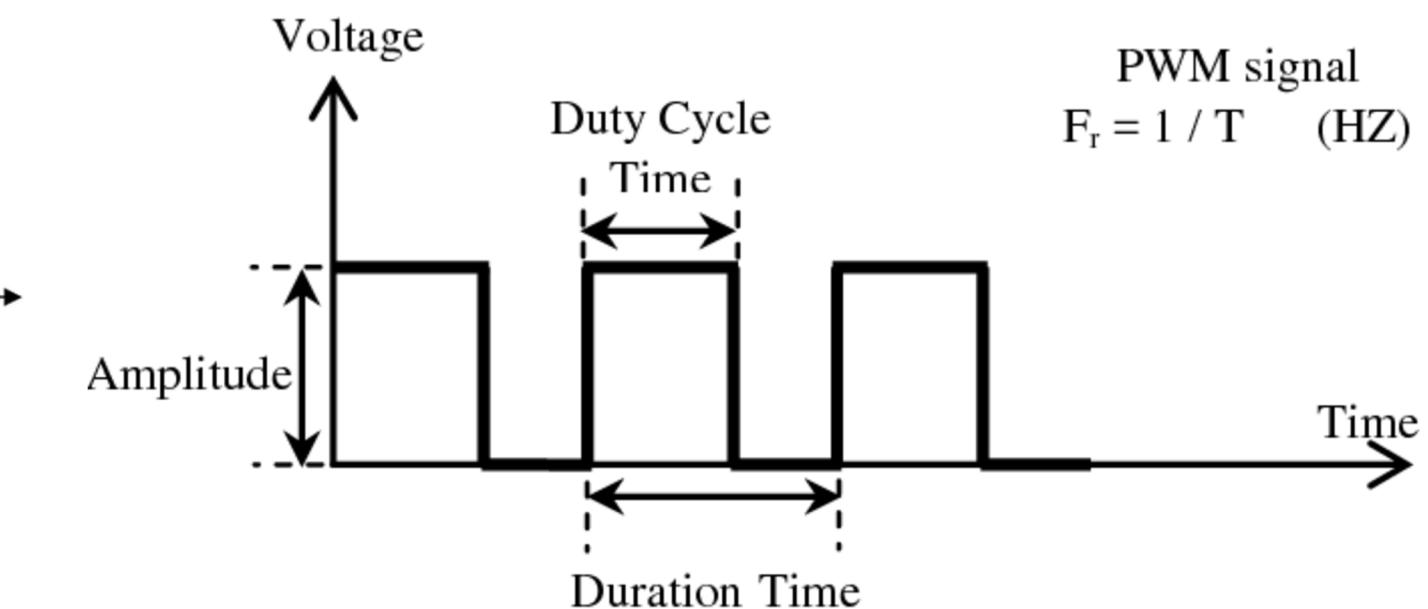
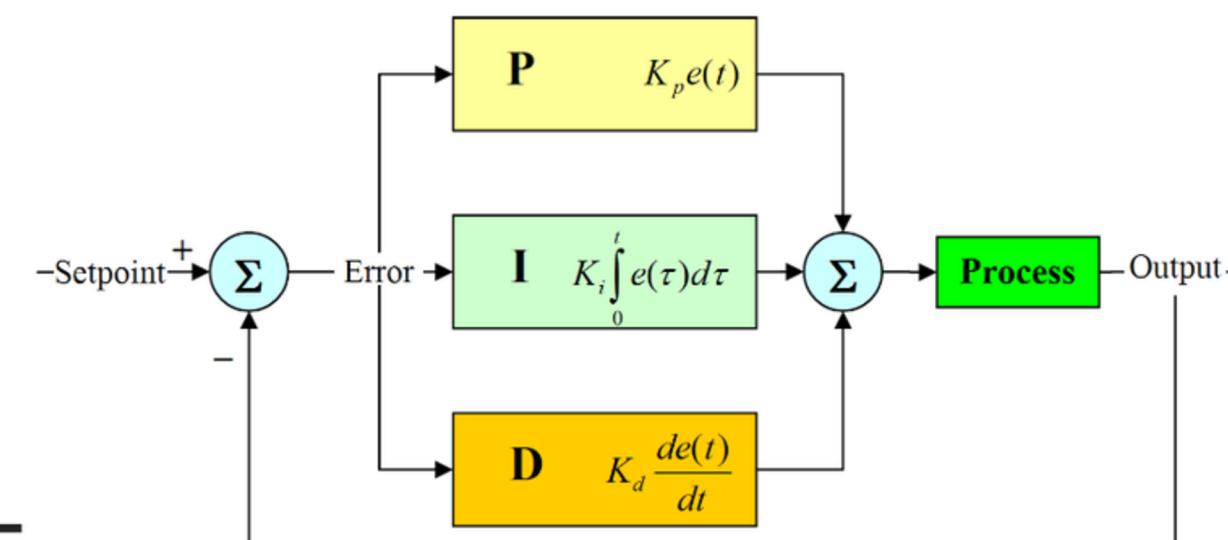
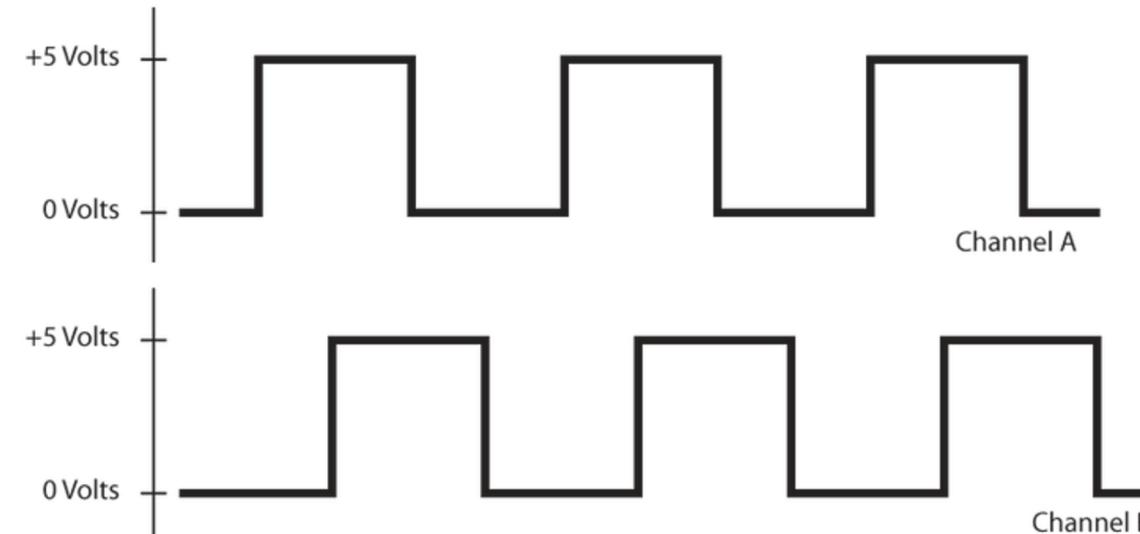
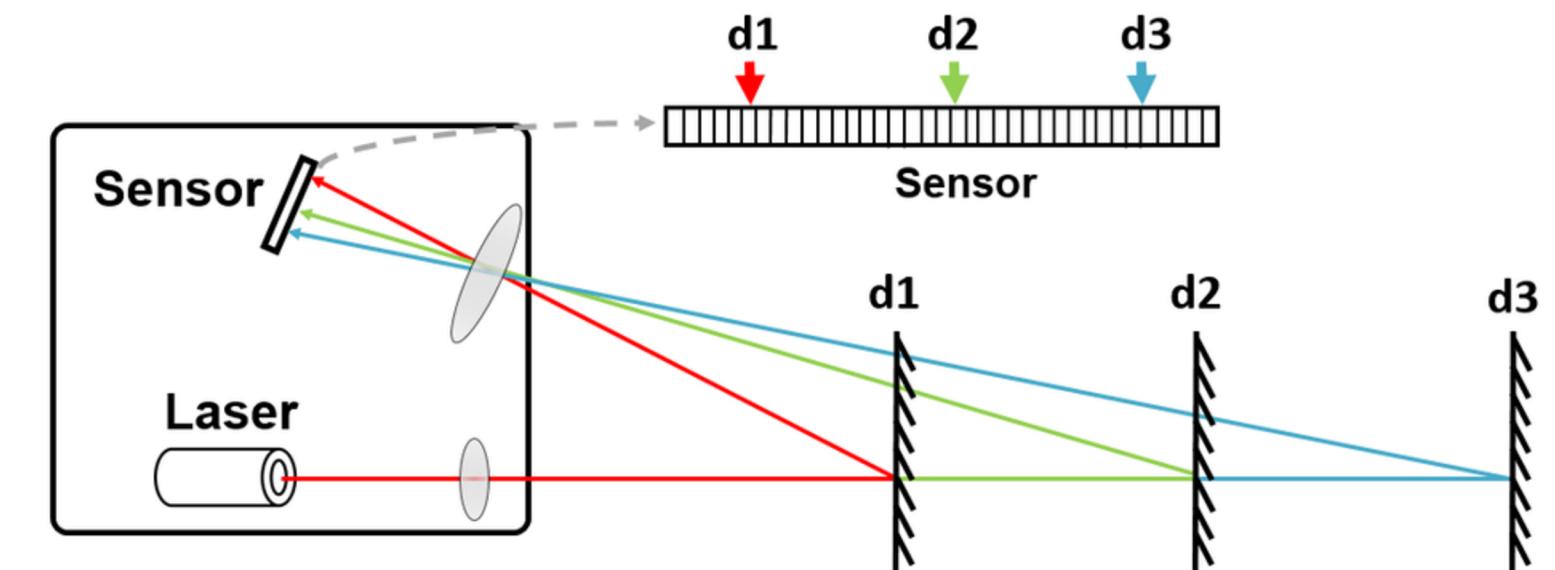
Actuators: 4 12V DC Motors, high-torque servo motor

Low-Level Controller: STM32F4 series MCU

High-Level Controller: Raspberry Pi 4

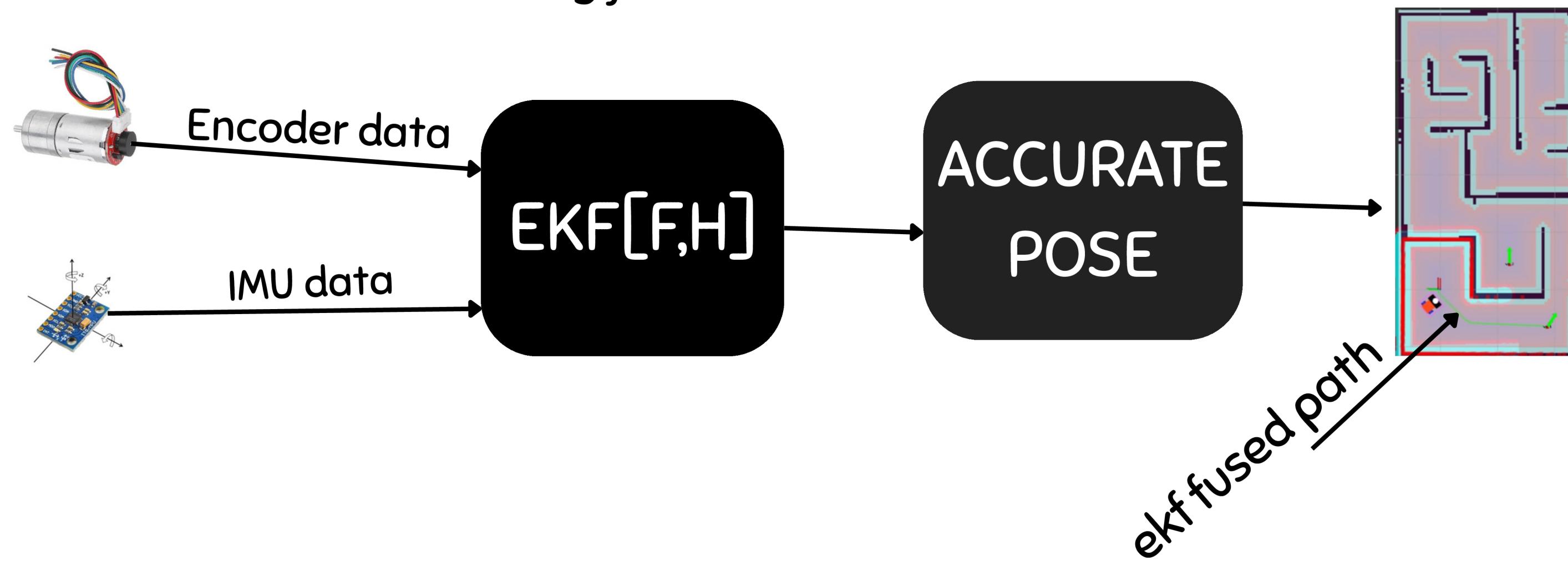
Power Supply: 11.1V Li-Po battery, USB Power bank

Featuring High Frequency PWM; PiD motor control



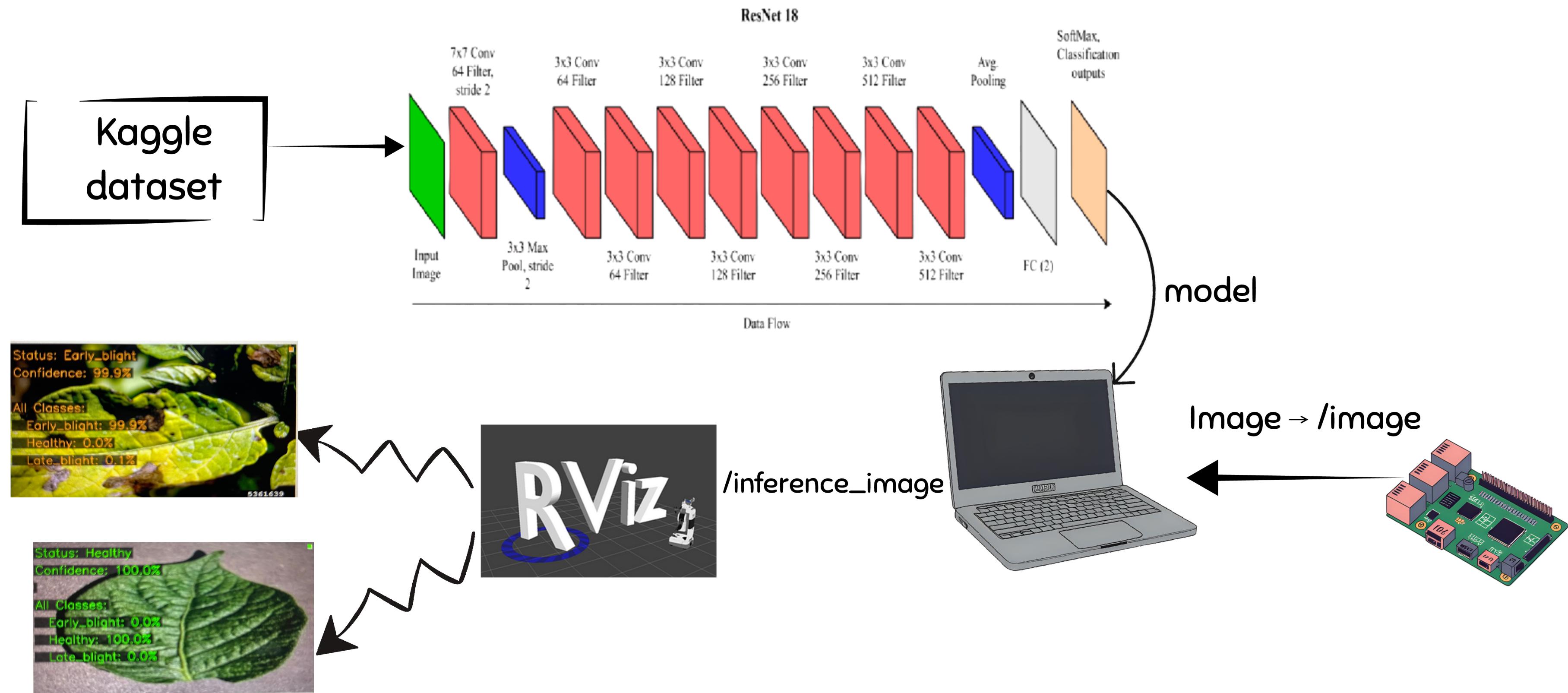
ACCURATE LOCALIZATION THROUGH SENSOR FUSION

- Encoders – Track wheel rotations → estimate linear & angular displacement
- IMU (MPU6050) – Measures acceleration & angular velocity → robot orientation
- EKF Fusion – Uses non linear models like robot motion and orientation and linearizes them around the current estimate using jacobians



COMPUTER VISION

IMAGE CLASSIFICATION



COMPUTER VISION

IMAGE CLASSIFICATION

Resnet 18 architecture used - 18 layers deep . Utilized transfer learning approach

Pretrained on 1.2 M ImageNet images - 1000 classes

Target - 3 classes (healthy, early blight, late blight) → in potatoes

Early blight - Dark brown spot with concentric rings

Late blight - Gray-Green water soaked lesions

Prepared a new dataset with 21452 images

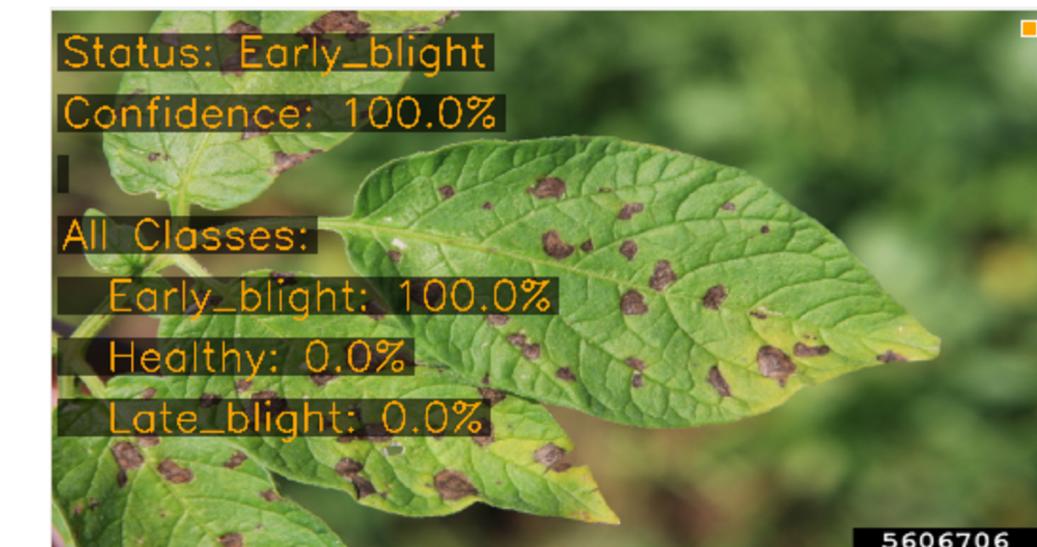
Why Resnet 18?

- Pretrained.
- Lightweight → 11 M parameters. Good for this task compared to Resnet 50+
- Proven effective for plant disease detection → don't reinvent the wheel
- Relatively less computational resources

Optimization algorithm → Adam

Why Adam?

- Good for small datasets
- Faster convergence than SGD.



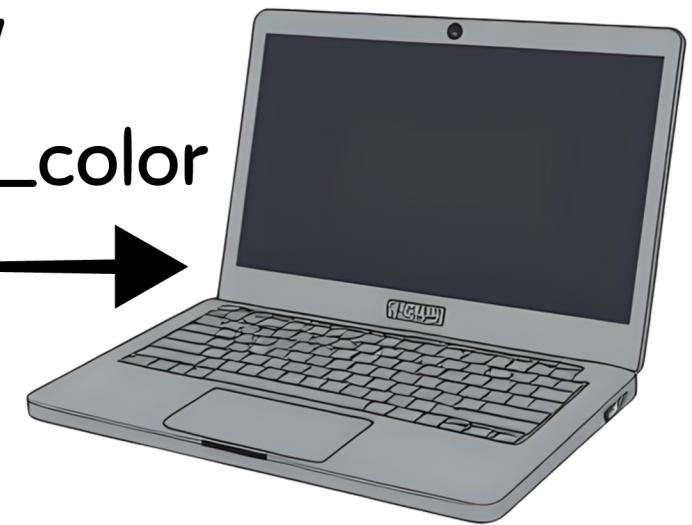
COMPUTER VISION

OBJECT DETECTION

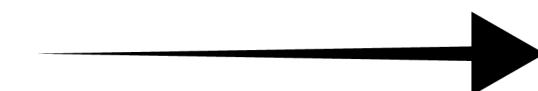
Task → Deposit the loaded cube to the correct color coded offloading area.



Image → /
detected_color



/servo_controller
/commands



For color detection – OpenCv extensive libraries

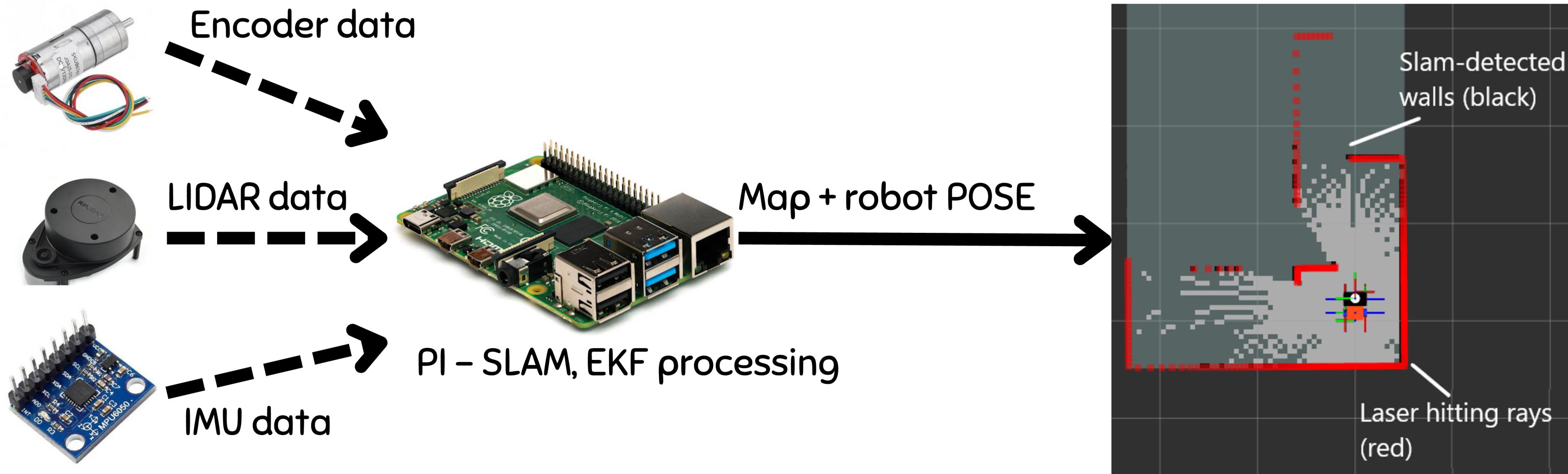
HSV color space preferred over RGB

Time based rebounding – real world robustness

Modularity achieved – ROS2 nodes (2)

SLAM AND NAVIGATION

- SLAM (Simultaneous Localization and Mapping) – Allows a robot to create a map of its environment while keeping track of its own location within the map.
- It uses data from sensors such as encoders, LIDAR and IMU and odometry to build a consistent map without losing the robot's POSE (Position and Orientation).



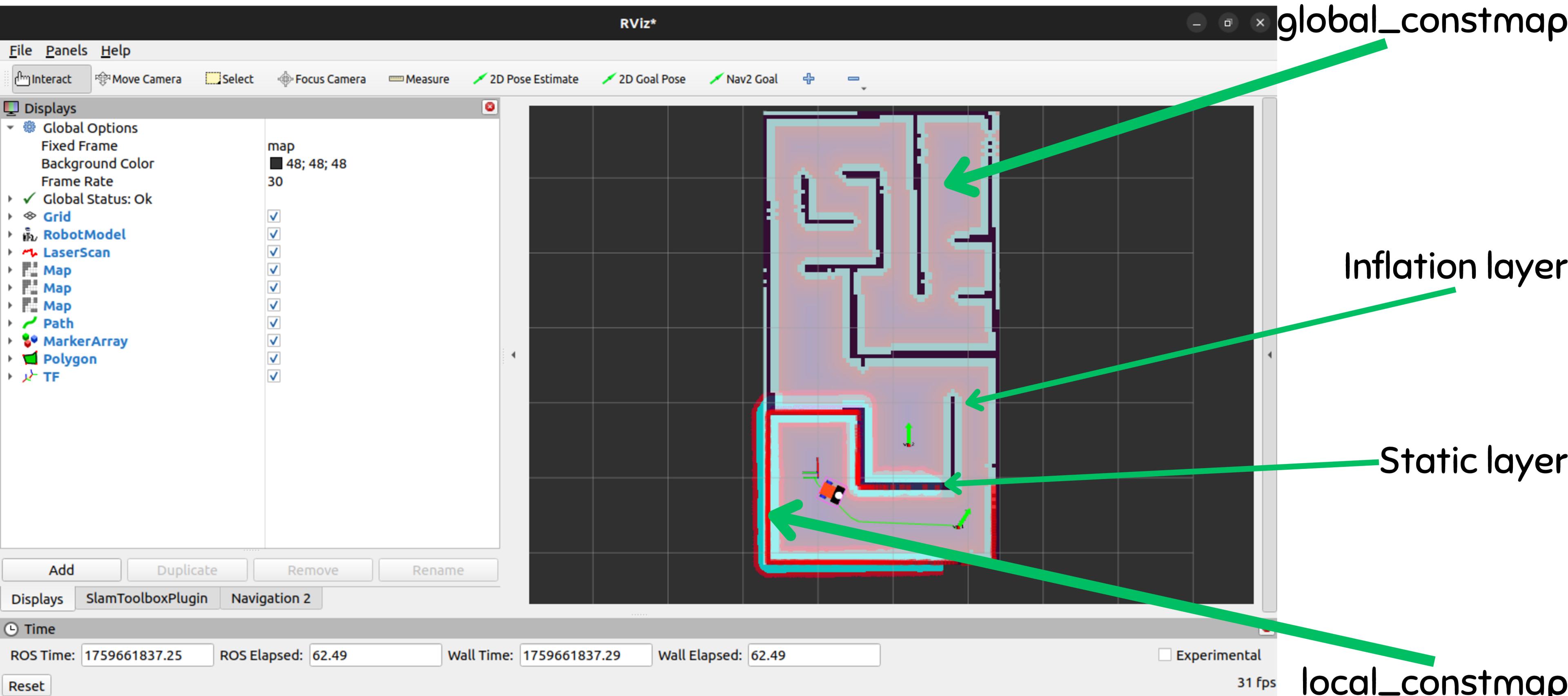
Some of the packages used in SLAM include:

- `slam_toolbox`: Responsible for publishing the coordinate transformations between the map frame and odom (odometry) frame which allows the robot to understand its POSE in the map.
- `amcl` (Adaptive Monte Carlo Localization): Used mainly after mapping has taken place and facilitates autonomous navigation of the robot within the map by using LIDAR data for localization.

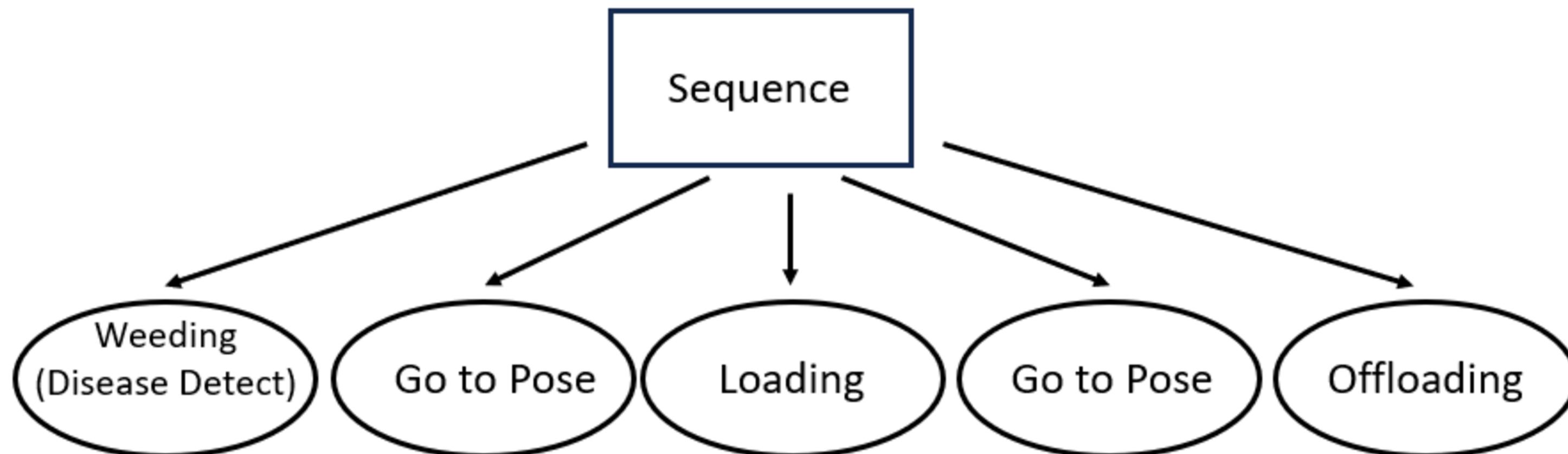
After mapping, Navigation takes place using a costmap. A costmap contains cells that quantify the difficulty of the robot traversing each cell.

The costmap is made up of two types:

- `global_costmap`: Covers the entire map and used for long distance path planning.
- `local_costmap`: Covers the immediate surroundings of the robot and helps the robot avoid new obstacles not recorded on the saved map.



PYTREES



THANK YOU