Institute Technical Summer Project 2023

DRONEacharya

Pranav Gupta, Shantanu Edgaonkar, Sarvesh Patil, Aditya Gaur | Ameya Marakarkandy

Introduction or Abstract

The project consists of a drone-rover system running in a mother-daughter type of a configuration. The quadrotor is manually controlled by a human user and is equipped with a camera module taking continuous footage of the model golf field below, which it relays to the ground station that detects the rover and golf balls and maps their positions on the field. The ground station then plans out an optimal path (minimum distance) made of straight line segments passing through all the golf balls for the rover on ground and relays the planned path to the rover. A MATLAB instance on the ground station takes all this information and uses it to intelligently drive the rover, ensuring through closed loop control that each golf ball is collected and returns back to a predefined home point.

Mission Detail

For the purpose of demonstration, there will be a small arena divided into a grid with corners indicated by ArUco markers. White table tennis balls will be scattered in this arena. We will take an image of this arena using a drone mounted camera. The balls will be detected using a pretrained object detection algorithm, and their coordinates with respect to their closest visible bottom left ArUco will be saved to a .csv file. These coordinates will then be sorted to find the optimal path which requires least time to traverse. The sorted coordinates will be fed into a MATLAB Differential Drive Pure Pursuit controller which will send the required wheel RPM instructions to the rover through ROS2. The rover will have an ArUco on top of it, its position will be found using the image from the camera mounted drone. Its wheel's RPMs will be controlled using a simple PID controller algorithm which will take input in the form of the live position of the rover. The rover will collect each ball with a rotating collection mechanism. The drone is currently manually controlled but later on could be programmed to maneuver field autonomously by localising itself wrt the ArUco markers.

Tools & Techniques Learned

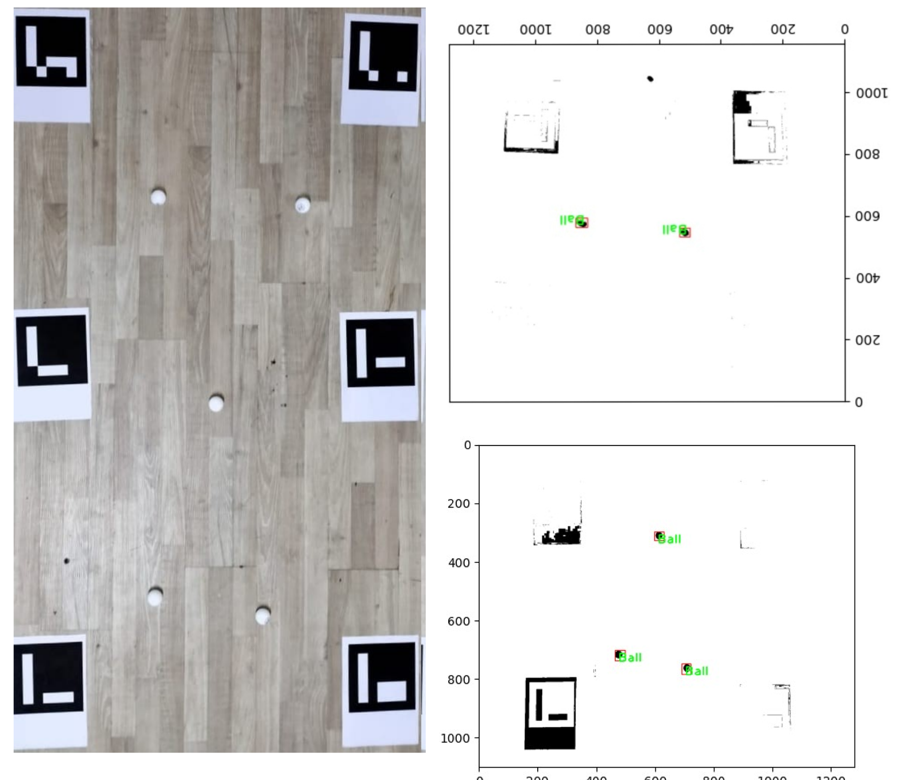
01. Raspberry Pi and libcamera to interface with camera
02. local webserver on RPi and publishing footage to it
03. bash shell, ssh, Raspberry Pi and Arduino libraries
04. Aruco markers for realtime localisation and mapping
05. Image manipulation using OpenCV2 and object detection using tensorflow SSD mobilnet V3 CNN algo
06. Differential Drive Pure Pursuit controller in MATLAB
07. Wireless communication among devices using ROS2
08. pyserial for serial comm with Arduino through USB
09. Encoder Motors with PID to maintain set RPM
10. Quadrotor dynamics, part picking, assembly and use
11. 3-D CAD modelling using Solidworks and Fusion 360
12. Multidomain System Modelling in Simulink.

Future Plan

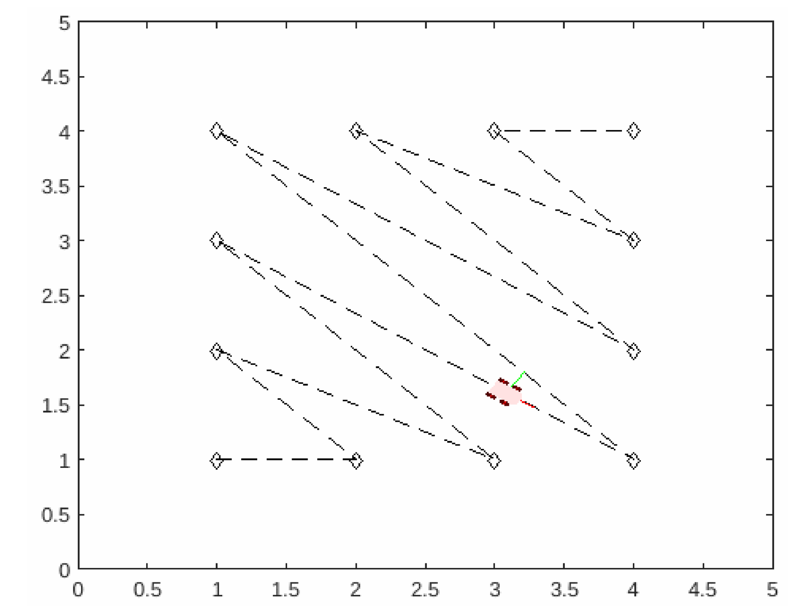
1. onboard computing functionality to the drone in order to eliminate dependence on a separate ground station.
2. smart controller friendly user dashboard system
3. Implementing AI so that drone and rover(s) can act as a complete autonomous system that just needs that mission parameters to intelligently traverse ground
4. docking station on the rover so that the drone can be mounted on the rover and hence both the machines act as a payload for each other as and when needed.
5. Extension of rover capabilities by deploying a swarm of rovers and implementing swarm algorithms on the rover swarm controlled with the help of a drones.
6. proper stable hardware mounts and housing for all mechanical and electrical components of the project.
7. deploying better quality and higher accuracy motors etc to improve finish of execution of autonomous task.

Results

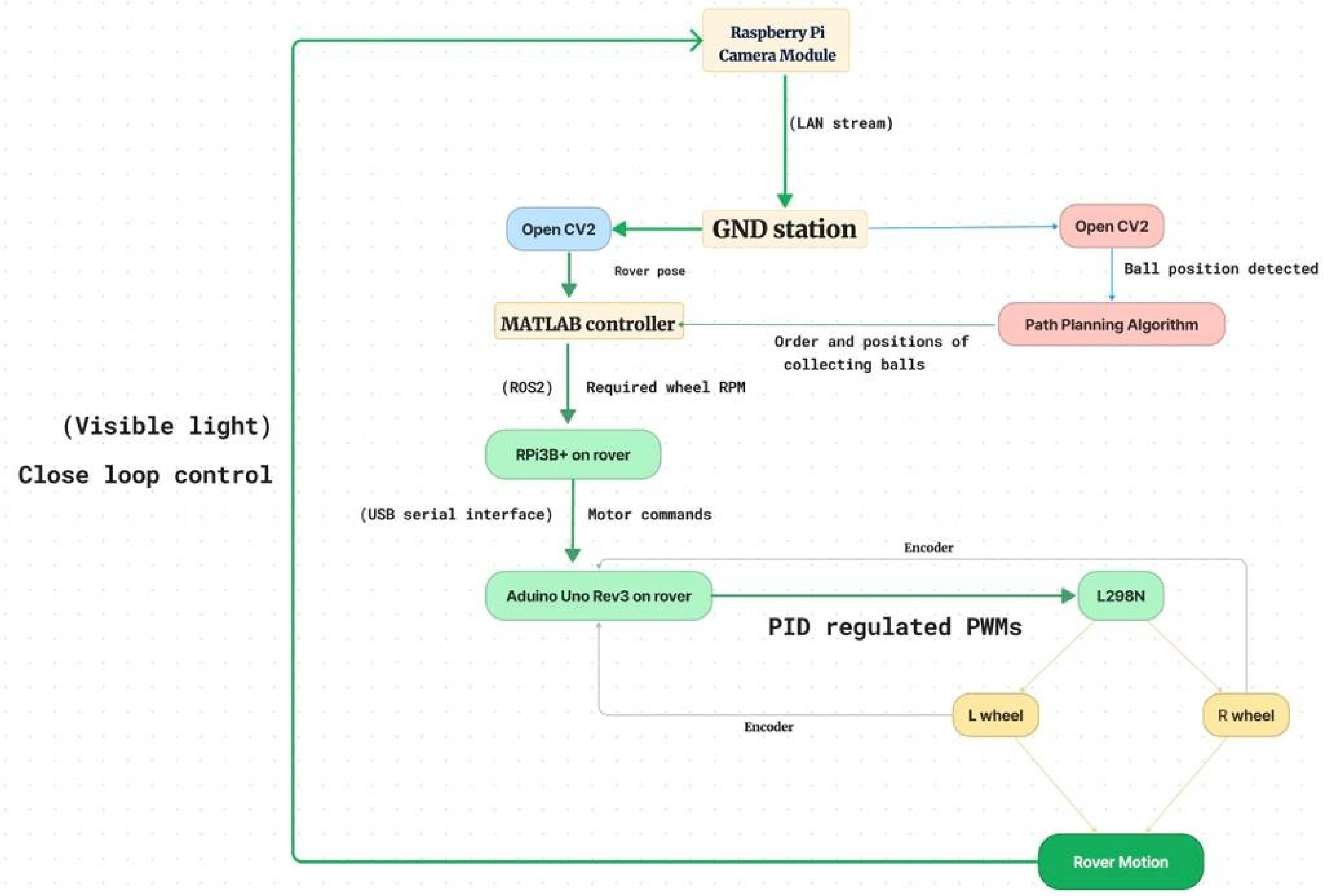
The following have been successfully implemented:-
0. fetching live low latency footage from raspicam wirelessly
1. detection of white balls and localisation wrt ArUco markers
2. optimization of traversal order to collect balls in least time
3. obtaining realtime position of rover for close loop control
4. publishing of wheel RPMs to RPi onboard the rover
5. serial communication from RPi to Arduino Uno to control motors
6. accurate motor speed control using PID on encoder readings
6. collection of balls in rover front basket by a rotating mechanism
7. real time coordination of tasks on ground station



detection of balls in low contrast



MATLAB simulation of rover



high level control flow diagram of mission execution