

Tethered Multi-Robot System for Improved Path Planning and Exploration

Callen Fisher¹ and Kyle Ahlschlager¹

Abstract—This paper presents a tethered unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV) for improved path planning and exploration in the presence of tall objects. The aim was to develop a UGV capable of mapping, planning a path, and exploring in a cluttered, unseen environment. Due to the low height of most UGVs, a tethered UAV was added to enable it to look over obstacles and improve its navigation and exploration algorithms. Having a tethered system avoids having the UAV perform complex algorithms like path planning and mapping, with simple altitude controllers serving the purpose of an elevated view point for the rover. This allows for off the shelf UAVs to be used. This paper describes the entire system, from a custom-built UGV in the form of an RHHex-style robot, to four point tether system, and the modifications made to an off-the-shelf UAV. Additionally, the software architecture, as well as the communication between the different systems was discussed in detail, particularly focusing on when the UAV should be deployed. The approach taken to integrate the UAV within the UGV was carefully designed to be platform independent allowing for more versatility in application. The platform was tested in an unknown environment with tall obstacles, herein the UGV was tested both with and without using the drone. The results showed an overall improvement in three key metrics, namely, reductions in both the time taken (-22%) and power consumption (-27%) required to locate and reach a set goal, and increased path-tracking accuracy (+38%).

I. INTRODUCTION

As industries slowly move towards automating processes and focus on improving safety for their workers, the importance of robust and efficient robotic platforms is becoming apparent [1]. One field that is rapidly progressing is that of inspection robots in hazardous environments [2]. Depending on the environment, these robotic systems take many forms, with a number of examples provided below. These robots are typically either fully autonomous or remotely operated (teleoperated) with autonomous subsystems [3]. These robots are sent into an area before humans proceed, to ensure their safety.

Several approaches have been taken to develop these inspection robots. For smooth environments, there are several wheeled platforms with a host of sensors that can autonomously explore an area or be controlled through teleoperation. A commonly used platform is the Clearpath Robotics Husky robot [4], along with robots from Superdroid Robots [5]. The downfall of a wheeled system is that it struggles to handle non-smooth terrain [6]. To solve this problem, some researchers have developed tracked vehicles that can

transition moderately rough terrain. One such example is the tracked surveillance robot HIRAD [7].



Fig. 1. On the left is the frame of the RHHex styled rover, with the final built rover on the right, with the tether and drone. Note the offset leg allowing for full rotation with overlap and preventing collisions.

Rough terrain continues to hinder tracked vehicles, this has resulted in the development and application of legged robots that are capable of traversing a multitude of terrains [8]. Great success has been shown with Boston Dynamics Spot [9], Ghost Robotics Vision 60 [10], and ANYbotics ANYmal robot [11], to name just a few. These robots easily transition rough terrain and can carry a small payload (sensor pack), however, these robots also have a few downfalls. Legged robots are incredibly expensive, typically they require much more complex controllers in order to maintain stability in motion and are often slower than their wheeled/tracked counterparts.

The RHHex platform [12] was designed to provide a good compromise between the rough terrain abilities of a legged robot and the low-cost aspects of a wheeled platform. Additionally, all of the solutions listed above are relatively short (most are under 1 m tall), resulting in standard objects occluding their vision systems, whereas humans could easily see over them. Due to this, even state-of-the-art ground-based platforms will struggle to perform path-planning [13], mapping (simultaneous localization and mapping, SLAM [14]), and exploration algorithms [15]. Tall objects in cluttered environments render these robots effectively blind, unless they have access to a global map, which is not the case in unseen environments.

One approach that avoids this problem is that of unmanned aerial vehicles (UAV). The literature has shown great success in mine tunnel mapping [16], search and rescue operations [17], and area exploration [18]. As these platforms are aerial, rough terrain is not an obstacle, and these UAVs can adjust their height according to the terrain. These platforms are limited by their flight time and load-carrying capacity (Elios 3 [19], a novel drone for tunnel exploration, has a flight time of 12.5 minutes without a payload). Additionally, some areas

*This work was funded by the NRF under grant number 129830

¹ Authors are with the Electrical and Electronic Engineering Faculty, Stellenbosch University, 7600 Stellenbosch, South Africa
cfisher@sun.ac.za

are classified as no-flight zones, especially during disasters [20], allowing aerial emergency resources the air space to deal with the hazards.

A novel approach that has been taken, is that of combining a ground-based vehicle and a tethered drone (tethered UAV or TUAV). One successfully deployed system was the Oxpecker (the TUAV) and Rhino (the UGV) [21]. This system made use of a wheeled robot with a self-levelling landing platform. The TUAV was used to scan pillars in underground mines that were up to 12 m high. Their focus was on a fully autonomous system due to the communication issues in underground mines. One limitation of drones is their flight time, therefore the Oxpecker made use of a powered tether, allowing the drone to run off the batteries in the UGV. This had the benefit of making the drone lighter. One of the downfalls was the power loss experienced in the cable, recorded at 123.5 W. Alternatively, other systems use the tether cable for communication [22], or a steel cable is used [23], allowing the drone to fly and hook the tether to a large object, allowing the rover to winch itself (this has been used to climb up small cliff faces).

Here we present the design of a low-cost system that consists of a ground-based rover (an RHex [12] styled platform) and a TUAV. Having a TUAV allows for the use of simpler UAVs that cannot perform path planning or mapping, and purely serve as an elevated view point for the rover. As the UAV can easily reach upwards of 3 metres, a telescopic mechanism was not considered. We are not focused on the performance of the rover or the drone, more the complete system, algorithms, and sensors. We are focused on developing a system that is platform agnostic, working on any form of UGV (wheeled, tracked, or legged) with any off-the-shelf or custom built UAV. Sections II and III detail the design of the system and integration of the UGV and TUAV with the various sensors, followed by the control of the ground-based rover in Section IV and the control of the drone and tether system in Section V. Following this, the path planning and exploration algorithms are investigated in Section VI and the adaptations for the drone camera in Section VII. The experimental setup is detailed in Section VIII with the results and discussion shown in Section IX. The paper ends with the conclusion and future work section presented in Section X.

II. SYSTEM DESIGN

The system consists of three main components. There is the RHex [12] style UGV, the UAV (a standard 5 inch quadcopter, Eachine Wizard X220 V3), and a tether system that connects the two robots. These components are described in greater detail below:

A. Rover Design

The ground-based robot was designed as an X-RHex [24] styled robot, which allowed it to traverse rougher terrain compared to wheeled-based platforms. This was required as the tether system reduced the overall reach of the drone, requiring the rover to have some rough terrain capabilities.

The frame of the robot was made from 6061 aluminium and consisted of two runners with three cross beams, all of 3 mm thickness. These were riveted together and contained slots and holes for easy cable management. To improve the rigidity of the structure, a 1 mm thick base plate was bolted to the frame. This also protected the bottom of the robot when traversing rough terrain. Additionally, “U-shaped” brackets were made for mounting the motors, 3 mm thick. The frame of the rover can be seen in Fig. 1, the base plate was excluded to provide a better view of the main structure.

The robot consisted of 6 Pololu 4753 50:1 brushed DC gear motors, each with a 64 counts per revolution (CPR) encoder attached. The middle motors were offset by 30 mm (sticking further out) from the front and back motors, allowing the legs to safely overlap and avoid collisions. These legs were required to support the full weight of the system. As such, the side profile was designed in CAD software, allowing for the legs to be laser cut from 6062 aluminium of 12 mm thickness, this ensured sufficiently thick legs for mounting and locomotion.

The mounting points on the legs was offset by a 45°angle ensuring that the centre of the legs were in line with the motor shaft, and the period of maximum force coincides with the standing range of the RHex platform. The couplers used to attach the legs and motors were designed in two pieces, the upper piece was a flat profiled rectangle, and the bottom had a circular slot machined in it corresponding to the motor shaft diameter. This design allowed for the couplers to clamp around the motor shaft and made use of the “D-shaped” shaft for power transfer, see Fig. 2.

Additionally, FEM calculations were done in CAD software to ensure the success of the part. The forces and torques used in the static stress analysis were based on the robot specifications and operation conditions. A load of 38 Newtons on the normal face of the leg and a maximum torque of 6.8 Newton-metre were used in this analysis. A solid mesh was used with a tolerance of 1.5 mm and an element size of 3 mm. This analysis concluded that the legs would be able to function adequately during system operation without suffering significant permanent deformation. The maximum deformation of the legs, was determined as less than 10 μm . The final rover, with tether and drone, can be seen in Fig. 1, and the rover’s physical parameters are shown in Table I.

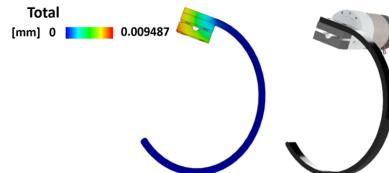


Fig. 2. On the left, the FEM analysis is shown. On the right, the bracket that bolts the leg to the motor shaft is shown. Note how the D on the shaft is used to ensure no motor slippage.

B. UAV Design

The UAV used for this project was the 5-inch Eachine Wizard X220 V3. Additionally, a custom built 3D printed

TABLE I
ROVER PARAMETERS

Parameter	Value
Dimensions	40 × 56 × 18 cm
Weight	6.8 kg
Leg Motors	50:1, 21 kg · cm
H-Bridge	MC3386 5A

pan/tilt mechanism, using two SG90 9g servo motors was attached to the front of the drone, allowing the camera to scan the area around the drone, as seen in Fig. 3.



Fig. 3. On the left, the drone with pan/tilt camera mount can be seen and on the right the drone connected to the RHex platform.

As the UAV was tethered, a simple height controller was implemented and the drone was not expected to perform any complex maneuvers. Additionally, the drone did not have to perform path planning or mapping, and was purely an elevated view for the rover.

C. Tether Design

The tether mechanism was designed to be simple and compact, allowing for it to be mounted within the rover, with the rover body providing structural support and protection from the environment. The tether system consisted of a single Pololu 2828 brushed DC motor with a 150:1 gear reduction and 64 CPR encoder attached. A GT2 belt system was attached to the motor and rotated two rods. These rods had bearings and attached to the rover frame via 3D printed brackets, as seen in Fig. 4. These brackets were placed 10 cm apart (centre to centre) and contained the spooled Trilene fishing line that was then attached to the drone. Therefore, by rotating the single motor, both rods rotated in the same direction, at the same speed, and either retracted the drone, or allowed slack, enabling the drone to launch into the air, to a maximum height of 3 metres. Fishing line was used as the tethering cable due to its reduced thickness to test (weight the line can support before breaking) ratio. The complete tether system can be seen in Fig. 4. For this system and application, the tether cable was purely to attach the drone to the rover, no communication signals or power were transmitted through the cables.

Due to the spooling points of the tether system being further apart than the attachment points on the drone, the cables provided increased stability to the drone, enabling it to perform stable hovering with simplified control algorithms. When the drone was fully retracted, the tether system ensured it remained safely secured on the robot and would not fall off when the rover was navigating rough terrain.

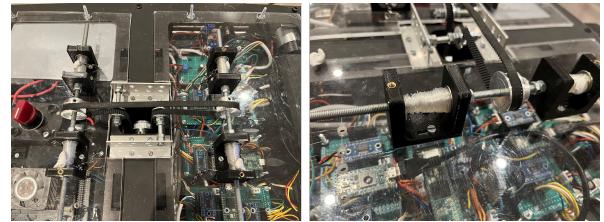


Fig. 4. On the left, the tether mechanism can be seen and on the right the spooled Trilene nylon fishing line.

III. FULL SYSTEM INTEGRATION, SENSORS, AND ELECTRONICS

The main processing unit of the system, located within the rover, was an Nvidia Jetson Nano (main processor). This unit performed all the image processing, SLAM, and path-planning algorithms. From these algorithms, commands were transmitted to the three sub-systems, namely the rover, tether system, and drone.

To control the rover, commands were transmitted to a Teensy 4.1 micro-controller (rover master controller) via UART protocol from the main processor. This micro-controller synchronized the motion between the six legs. To do this, commands were further transmitted to two more slave micro-controllers, Teensy 4.0s (motion controllers). These each controlled three of the leg motors (due to the number of GPIO pins each motor required, multiple micro-controllers were required). These motion controllers interfaced with the H-Bridges (MC33886) controlling the leg motors, as well as reading the encoder values and transmitting the data back to the rover master controller. The master and motion controllers used both I2C and UART communications to function. I2C was used to change the state of the controllers as it allowed for a general call to all controllers on the bus and for controlling individual motors during calibration. UART communication was used during standard operation and used the FrSky Sbus protocol [25] to control the speed and position of each motor as it allowed for faster transmissions.

For control of the tether system, the main processor sent commands to the rover master controller mentioned above. This micro-controller interfaced with the H-Bridge (L298) controlling the tether motor and recorded its encoder position. The encoder on the winch motor was used to determine the approximate altitude of the drone and ensure the drone was not damaged due to over-retraction of the tether lines.

To control the drone, a Bluetooth Low Energy (BLE) link between the rover master controller and the drone was established, a JDY-23 BLE module was connected to the rover master controller over a UART connection and an ESP32 micro-controller (drone controller) mounted to the drone was configured as a BLE client to communicate with the rover master controller that acted as a server to transmit commands. The drone controller, was connected to the SBUS receiver port of the F405 flight controller of the drone, it

was configured to transmit an SBus signal over one of its Serial ports allowing it to control the drone by manipulating the SBus signals much as an operator would by using a transmitter. This configuration allowed for the rover master controller to control the drone. A live video feed from the camera on board the drone, was transmitted to the main processor for path planning and exploration purposes.

To achieve accurate positional and speed control of each motor onboard the rover, each motor had a 64CPR encoder. For this purpose, the encoders were required to have a uniform starting position and current sensors (Pololu ACHS-7122 bidirectional sensors) were connected to the rover master controller to perform consistent initialisation and control of the rover's legs. For the rover to perceive its environment, two types of sensors were used, namely, a LiDAR module (OKdo LD06) for distance measurements of the surrounding area and two IMX219 cameras, with a 77° field-of-view. The LiDAR module interfaced over a UART connection and provided a full 360° distance scan. These sensors were attached to the rover and not the drone, as the rover was considered far more robust than the drone.

To power the different sub-systems, two separate battery packs were used. The first was a 3S2P pack of 21700 Li-ion batteries, this powered the rover, tether system, and main processing unit. The second battery pack was a 4S 3300 mAh LiPo battery pack, which was mounted on the drone. It was decided to have separate batteries, as a powered tether system would increase the complexity of the tether, as well as result in large power losses due to the current draw and length of wires [21]. Additionally, the current requirement for powering the six RHx motors and the four drone motors was excessive to power from one portable battery pack. Each leg motor had a stall current of 5.5 A resulting in a possible peak current draw exceeding 66 A. The current battery packs enabled the rover to operate for a measured 45 minutes and the drone to fly for 16 minutes. The original RHx platform was able to stand for 48 minutes and continuous walking for 18 minutes. As this was the first prototype, this was considered acceptable. Future upgrades can be made to the rover battery to allow for extended running and a docking station can be setup to enable the drone to charge from the rover battery when fully retracted and landed on the rover.

The development cost of the entire system is shown in Table II. The total cost of the system was \$1800 which is considerably less than many of the currently available platforms.

IV. ROVER CONTROL

The main processor executes various exploration and path planning algorithms detailed in Section VI and high-level commands to the rover master controller. These commands were broken down into a direction and a relative speed that the robot should maintain (experimental results showed that the rover could move at a conservative 2.3 body lengths per second and a turning radius of approximately 5 body lengths while in motion), rotation in place, either clockwise or counter-clockwise, and deploying, extending the altitude and

TABLE II
COST OF DEVELOPMENT OF THE COMPLETE SYSTEM

Item	Quantity	Unit Cost (USD)
Aluminium Frame and Legs	NA	86
Motor (Pololu 4753)	6	288
Motor (Pololu 2828)	1	50
H-Bridge (MC33886)	6	65
H-Bridge (L298N)	1	3
Pan Tilt mechanism	1	13
3D printing	NA	500
Batteries (21700)	6	29
BMS 60A for LiPos	1	9
Jetson Nano B01	1	180
Teensy 4.1	1	32
Teensy 4.0	2	48
IMX219-77	2	37
LiDAR (OKdo LD06)	1	105
Drone (Eachine Wizard X220 V3)	1	280
Sundry Costs	NA	75
Total		1800

recalling the drone. The rover master controller transmitted positional data of each leg as well as the battery voltage and current draw at regular intervals to the main processor allowing for informed movement decisions to be made based on the available data.

The rover master controller generated a gait pattern to execute the commands received by the main processor, and transmitted the relevant speed and positions from the gait generator to the motion controllers. The motion controllers utilised a positional PID controller. Each motion controller had three motors and their encoders attached. These three motors formed a triangle or tripod. During the motion of the RHx there was always, at minimum, one tripod in contact with the ground, allowing the platform to be dynamically stable without requiring complex stabilisation controllers, as seen in many other legged robots [26]. For this project, only one motion gait was implemented, this was the tripod gait [27]. The tripod gait is a general purpose gait that provides a sufficient trade-off between stability and speed for most purposes. It is characterised by a slow-moving "stance" phase and a fast-moving "flight" phase. During the stance phase, the legs of a tripod are in contact with the ground and provide the propulsion off the ground required to move forward. During the flight phase, the legs of a tripod are moving quickly through the air to return to the ground contact point before the start of their next stance phase.

A simple clock-based system was implemented in order to control the motor's position, speed, and respective phase. By varying the combined time each tripod spent in the two phases, the speed of the rover was adjusted, and by varying the period of the stance phase turning was achieved. By increasing the stance period of one tripod, its motion would be slowed enough that the rover would slowly turn in the direction of the slower tripod. This is very similar to the skid steering implemented in tracked vehicles, albeit the velocity ratio between the two tripods must be carefully chosen to maintain the constraint of at least one tripod being in the stance phase at a given time. This approach to turning proved effective, but can be further improved by varying the range

of the stance phase and reversing the direction of rotation of a tripod instead of just slowing it to create a more extreme turning manoeuvre as seen when rotating in place.

Before any of the above can occur, the position of the motors and legs needs to be initialized, as the motor encoders are relative and not absolute. The rover master controller had three current sensors attached, one per pair of alternating legs, allowing for the current of a motor in its stance phase to be more accurately measured as the other connected motor would be in the flight phase and have a much lower current requirement at that time. For the initial calibration of the leg positions, the rover was placed on its belly, with the legs off the ground. One by one, the legs were slowly rotated until a current spike was observed. This indicated that the leg has touched the ground, the motor was then stopped, the encoder initialised to a known value, and the next motor calibrated.

V. DRONE AND TETHER CONTROL

The drone and tether system were two separate systems, however, they were interlinked when control commands were sent and form part of a state-machine controller. The drone would only be deployed when necessary, if the rover could identify a path to follow or see the goal, then the drone would not be deployed.

In the case that the drone needed to be deployed, the main processor would transmit a deployment command to the rover master controller, which in turn would transmit a corresponding command to the drone controller. The drone would subsequently arm, and the throttle values gradually increased until it was ready to take off. During this time the tether motor would unwind, creating slack on the tethers and allowing the drone to rise. The drone was set to reach an initial altitude of 1 metre where the first round of scanning would take place. The camera feed was streamed over a 5.8 GHz connection to a USB video receiver on the main processor. If the goal was located then the main processor would transmit a recall command to the rover master controller and the reverse operation ensued or an extension command was transmitted and the drone was allowed to rise even higher. The drones maximum height was limited to 4 metres to account for the testing areas limitations.

The control of the tether and drone was implemented by a finite state machine, where there were three main states, namely, “Landed”, “Deployed” and “Extending”, as well as some transitional states that handled momentary tasks such as arming and disarming of the drone. During the landed state the drone was still operational and its camera was moved to a position where it could see in front of the rover at a greater height than the two IMX219 cameras, which allowed for additional vision when performing ground-based scanning.

VI. ROVER PATH PLANNING AND MAPPING

During the UGV searching phase, the drone was landed, but its camera was still available for use in trying to locate the goal. This allowed for three cameras to be used for goal locating. An exploration algorithm was implemented in this phase that focused on rotating the rover to create a 360° view

of the surrounding area. During this time the cameras were searching for the set goal and the LiDAR unit was mapping the area to determine what paths were viable for traversal.

If the goal was located by the cameras, the rover would stop rotating, and the pixel coordinate of the goal’s location be recorded and used to determine where on the current LiDAR scan the goal was and its relative distance to the rover. This was done by first determining the distance “d” from the rover to the detected goal as shown in (1). Here B represents the distance in centimetres between the two cameras, f is the focal length of the cameras in the same units as B, and δ is the disparity between the two images. The disparity is given as the difference between the left and right “x” coordinates captured by the stereo cameras. The “y” coordinate are the same due to the cameras being in parallel.

$$d = (B \times f) \div \delta \quad (1)$$

Where : $\delta = x_1 - x_2$

The distance to the goal object was then used to determine its 3D position and use this in the LiDAR-based grid map for path planning. (2) shows how the pixel coordinates of the goal detected in the left image are converted to a 3D space representation of the goal object. Here f is the focal length in pixels.

$$X = (x_1 \times d) \div f \quad (2)$$

$$Y = (y_1 \times d) \div f$$

The A* search algorithm [29], was then implemented on the LiDAR data to the determined goal location and a path to goal is calculated and followed without the need for deploying the drone. A*, was used as it computes the shortest path to a goal location by default by selecting the path that minimizes (3). Here $g(n)$ represents the cost of the path from the start node to the current node (n) and, $h(n)$ is a heuristic function that uses the Manhattan distance to determine the distance from the current node to the goal node.

$$f(n) = g(n) + h(n) \quad (3)$$

The algorithm implementation is shown in Fig. 5, this runs on the main processor and was executed on initial startup and when there was no movement data available. If the goal was not found and a full rotation had been completed then the rover would prepare to launch the drone and implement the drone exploration algorithm discussed in Section VII.

VII. ADAPTATIONS FOR DRONE DEPLOYMENT

When it was determined that the rover could not locate the goal using the ground-based search in Fig. 5, or there were multiple potential paths that the rover was unable to determine where they lead, then the drone would be prepared for launch as described in Section V. The conditions for drone launch are narrowed down to the following:

- 1) After scanning the surrounding area, the goal was not visible to the ground cameras.

- 2) The LiDAR data revealed there were multiple paths to choose from.
- 3) The surrounding area was obstructed by large numbers of obstacles, rendering the area hazardous to pass

If any or a combination of the above conditions were met then it was determined to be necessary to deploy the drone to gain a deeper understanding of the surrounding environment. When the drone has been deployed, the camera feed streaming to the main processor used an object detection model (MobileNet-v2 [28]) that searched for the goal as the camera on the drone swepted over the area. If the goal was not located, the drone was instructed to rise higher and to scan again, if the goal was still not located once the drone had finished scanning at its maximum altitude then it was recalled and the rover instructed to select an arbitrary location to travel to, much like it would in the absence of the drone.

In the case that the drone detected the goal then this image was captured and the goal location in pixel coordinates gathered from the detection model, this could be used for selecting a goal location on the local path planner to travel closer to the goal. Additionally, before using the local planner, the aerial image was processed to convert the image pixel intensity data into clusters of potential obstacles. Homography was used for perspective removal and to create a representative aerial view. After this, the image could be broken down into a grid and the A* algorithm [29] was used to determine a potential path to the goal. The accuracy of this path was limited due to the use of a single camera, variable angle of the drone and distortion of the camera, but it could be used to influence the direction and measure the accuracy of the local path from the LiDAR data. The adapted algorithm implemented is shown in Fig. 5. This approach of using both the camera and LiDAR data to determine the most suitable path toward the goal proved to be effective as seen in Section IX.

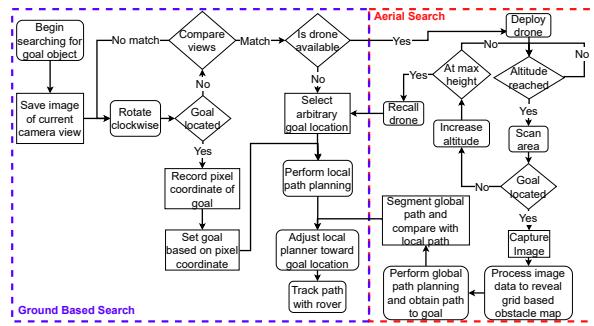


Fig. 5. The adapted exploration algorithm implemented on the rover for ground and aerial searching.

VIII. EXPERIMENT SETUP

The experiment layout was the same for the “rover only” and “rover with drone” experiments. The testing took place in a 10x10 metre area, populated with various obstacles. Each obstacle was chosen for its varying height, but it was required that none of the obstacles be low enough that, when standing the IMX219 cameras could see over them. Although, two

obstacles were used such that when landed, the drone camera would be able to see over them whilst the rover was walking. An ArUco marker cube was chosen as the goal obstacle. For both instances of the experiment, five tests were run and the results were recorded, the procedure for both was consistent. The rover and goal object were located in opposite corners but with a maze in-between blocking the view of the rover. The experimental area was simulated offline and an optimal path was generated, this path was used to determine the deviation of the rover from the ideal route in each test instance. The experiment layout can be seen in Fig. 6, this layout was captured by the LiDAR sensor onboard the rover.

IX. RESULTS AND DISCUSSION

This section shows the results from the experiments where the system was evaluated both with and without the use of the drone providing the aerial view.

A. Rover Results

Fig. 6 shows the paths chosen and tracked by the rover over the five testing instances without the drone being used. Each circle represents where the rover stopped and implemented the ground-based searching algorithm detailed in Section VI, note that a scan is performed by default at the start of the test as the goal is not visible.

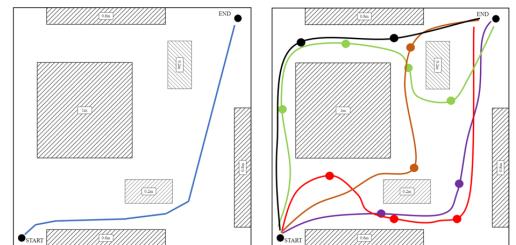


Fig. 6. Paths chosen by the rover when implementing a ground-based search, note how this differ significantly to the optimal path.

B. Rover with Deployed Drone Results

Fig. 7 shows the paths chosen and tracked by the rover over the five testing instances while using the drone to provide an aerial view. Each circle represents the location where the drone was deployed and the adapted search algorithm described in Section VII was implemented, note that an aerial scan is performed at the start of the test as the goal is not visible without use of the drone.

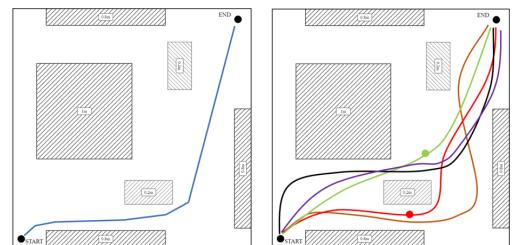


Fig. 7. Paths chosen by the rover using the drones aerial view, note how these closely follow the optimal path. Two testing instances needed a 2nd drone deployment, herein the rover needed to relocate the goal due to deviating from its path.

C. Combined Results

Fig. 8 shows the physical parameters that were measured when conducting each experiment. The results shown include the voltage drop measured from when each test started to its conclusion, the time taken to conclude each test, the number of failures that occurred (these included inability to locate the goal and physical failures) and the number of scanning stops made by the rover (for the tests with the drone this is both stops to deploy the drone and stops for only ground-based scanning). Additionally, a deviation from the optimal path is given.

The average voltage drop recorded during testing with the drone active was 1.12 V compared to the 1.54 V without, this represents a decrease in power consumption of 0.42V or 27% decrease in consumed power. Similarly, for the time taken to reach the goal, for the testing with drone an average time of 300s and 385s without. This represents an improvement of 85s or 22%. The deviation from the optimal path also decreased when using the drone, 13% when using the drone compared to 33% without, representing a 38% improvement.

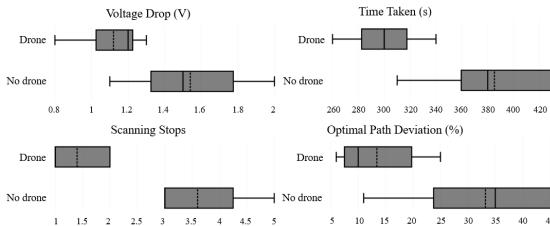


Fig. 8. Comparison of the key metrics used to determine the rovers performance between the tests with and without use of the drone, note the uniform improvement achieved when using the drone.

X. CONCLUSION

This research presented a low-cost system that was shown to autonomously deploy a tethered drone to assist in navigating unseen environments to locate a goal position. Overall, we found that deploying the drone increased the exploration algorithms efficiency and biased the rovers motion towards the goal location. Future work will involve testing this system in harsh environments, such as underground mines, as well as improving the drone control. Additionally, a charging station for the drone will be mounted on the rover, allowing the drone to charge while stationary on the rover.

ACKNOWLEDGEMENT

The authors would like to thank the NRF for funding this research under grant number 129830.

REFERENCES

- [1] Noralfishah binti Sulaiman, et al. "The Role of Autonomous Robots in Fourth Industrial Revolution (4IR) as an Approach of Sustainable Development Goals (SDG9): Industry, Innovation and Infrastructure in Handling the Effect of COVID-19 Outbreak." IOP Conf. Ser: Earth Environ. Sci (2021): 775.
- [2] Trevelyan, James, Kang, Sungchul, and Hamel, William. "Robotics in Hazardous Applications." Springer Handbook of Robotics (2008): 1101-1126.
- [3] Astrid Weiss, et al. "Autonomous vs. tele-operated: how people perceive human-robot collaboration with hrp-2." In Proceedings of the 4th ACM/IEEE international conference on Human robot interaction (2009): 257-258.
- [4] "HUSKY: UNMANNED GROUND VEHICLE." 2023 accessed 20 July, <https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/>
- [5] "ROBOTS THAT SOLVE BUSINESS CHALLENGES" 2023 accessed 20 July, <https://www.superdroidrobots.com/>
- [6] Todd, David J. "Walking machines: an introduction to legged robots." Springer Science & Business Media, 2013.
- [7] Mohebbi, Abolfazl, et al. "Design, simulation and manufacturing of a tracked surveillance unmanned ground vehicle." 2010 IEEE International Conference on Robotics and Biomimetics. IEEE, 2010.
- [8] J. Hodgins, "Legged robots on rough terrain: experiments in adjusting step length." Proceedings. 1988 IEEE International Conference on Robotics and Automation (1988): 824-826 vol.2
- [9] Moses, J., Ford, G. "See Spot save lives: fear, humanitarianism, and war in the development of robot quadrupeds." Digi War 2, 64-76 (2021).
- [10] "Ghost Robotics: Vision 60." 2023 accessed 21 July, <https://www.ghostrobotics.io/vision-60>
- [11] "Meet ANYmal, your new inspector." 2023 accessed 20 July, <https://www.anybotics.com/anymal-autonomous-legged-robot/>
- [12] Saranli, Uluc, et al. "RHex: A Simple and Highly Mobile Hexapod Robot." Departmental Papers ESE (2001): 20.
- [13] Y. Wu, et al. "Cooperative Path Planning of UAVs & UGVs for a Persistent Surveillance Task in Urban Environments." in IEEE Internet of Things Journal, vol. 8, no. 6 (2021) 4906-4919.
- [14] Frese, Udo, et al. "A SLAM overview from a users perspective." KI. (2010) 24. 191-198.
- [15] Narayan, S., et al. "A priority based exploration algorithm for path planning of an unmanned ground vehicle." 2014 International Conference on Embedded Systems (2014): 275-280.
- [16] Shahmoradi, Javad, et al. "A comprehensive review of applications of drone technology in the mining industry." Drones 4.3 (2020): 34.
- [17] Mishra, Balmukund, et al. "Drone-surveillance for search and rescue in natural disaster." Computer Communications 156 (2020): 1-10.
- [18] Park, S., Choi, Y. "Applications of Unmanned Aerial Vehicles in Mining from Exploration to Reclamation: A Review." Minerals (2020), 10, 663.
- [19] "Elios 3: Digitizing the inaccessible." 2023 accessed 24 July, <https://www.flyability.com/elios-3>
- [20] "Drone Laws: For a Safer Airspace." 2022 accessed 28 June, <https://drone-laws.com/drone-laws-in-south-africa/>
- [21] Martinez Rocamora Jr, Bernardo, et al. "Oxpecker: A tethered uav for inspection of stone-mine pillars." Drones, vol. 7, no. 2, 2023.
- [22] Fagiano, Lorenzo. "Systems of tethered multicopters: modeling and control design." IFAC-PapersOnLine 50.1 (2017): 4610-4615.
- [23] Miki, Takahiro, Petr Khrapchenkov, and Koichi Hori. "UAV/UGV autonomous cooperation: UAV assists UGV to climb a cliff by attaching a tether." 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.
- [24] Galloway, Kevin C. et al. "X-RHex: A Highly Mobile Hexapedal Robot for Sensorimotor Tasks." (2010).
- [25] "FPV Protocols Explained (CRSF, SBUS, DSHOT, ACCST, PPM, PWM and more)." 2023 accessed 13 March, <https://oscarliang.com/rc-protocols/>
- [26] Wieber, PB., Tedrake, R., Kuindersma, S. (2016). "Modeling and Control of Legged Robots." In: Siciliano, B., Khatib, O. (eds) Springer Handbook of Robotics. Springer Handbooks. Springer, Cham.
- [27] Haynes, G.C., Cohen, F.R., Koditschek, D.E. (2011). "Gait Transitions for Quasi-static Hexapedal Locomotion on Level Ground." In: Pradalier, C., Siegwart, R., Hirzinger, G. (eds) Robotics Research. Springer Tracts in Advanced Robotics, vol 70. Springer, Berlin, Heidelberg.
- [28] Chiu, Yu-Chen, et al. (2020). "MobileNet-SSDV2: An Improved Object Detection Model for Embedded Systems." 1-5. 10.1109/IC-SSE50014.2020.9219319.
- [29] Zammit, C, and van Kampen, E-J. "Comparison between A* and RRT Algorithms for UAV Path Planning." In Proceedings of the 2018 AIAA Guidance, Navigation, and Control Conference (2018).