MATEC Web of Conferences **388**, 04005 (2023) https://doi.org/10.1051/matecconf/202338804005
2023 RAPDASA-RobMech-PRASA-AMI Conference

Design of HERMES: a mobile autonomous surveillance robot for security patrol

John Dickens[1*], *Thabisa* Maweni[1], *Tiro* Setati[1] and *Zubair* Suddoo[1]

[1]CSIR Centre for Robotics and Future Production, Pretoria, South Africa

Abstract. The HERMES autonomous surveillance robot platform is a low-cost outdoor autonomous surveillance vehicle. It is designed to autonomously patrol outdoor areas performing surveillance and providing automated alerts of detected vehicles and people. The design and testing of this system are covered in this paper. The design philosophy focused on the use of off-the-shelf components wherever possible with the base of the robot being a modified electric quadbike. The testing has verified that the surveillance robot can perform real-time person and vehicle detection, video streaming, manual and autonomous navigation on a low-cost platform. The development of the robot platform is continuing with the current focus being on the improvement of the autonomous navigation, ingress protection (IP) rating and verification of the battery life.

1 Introduction

South Africa has the third-highest crime rate in the world [1]. There is an obvious need for a robust and efficient surveillance system that can monitor and detect any suspicious activities in real-time. Cable theft is an ongoing problem across the country and is costing the economy millions of rands. Transnet alone loose tens of kilometres of cable per week [2]. The distributed nature of electricity infrastructure makes surveillance challenging. An autonomous mobile surveillance robot can provide several advantages in this regard. Firstly, it can operate 24/7 without fatigue, unlike human security personnel. Secondly, it can patrol a large area in a relatively short amount of time, providing comprehensive surveillance coverage. Thirdly, it can reduce the risk to security personnel by allowing threats to be evaluated and prepared for. Overall, an autonomous mobile surveillance robot can be a valuable tool for enhancing security of both public and private property.

To be a useful tool for security surveillance, the cost of the robot platform needs to be kept low. Therefore, to reduce the cost of the platform, the design is built around a small electric quadbike with added sensors and actuators required for autonomous operation. To give the mobile robot a unique identifier, the name HERMES was chosen, which is an acronym for Highly Efficient Robotic Mobile Enforcement System.

The test results evaluate the performance of the platform in respect to the requirements that are outlined in the design section. The major research contributions of the paper are the following:

- To assess the performance of HERMES in terms of its speed, coverage area, and object detection ability during surveillance tasks.

---

[*] Corresponding author: jdickens@csir.co.za

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

- To investigate methods to keep the platform's cost low without compromising its efficacy.
- To design and implement the necessary sensors and actuators for the autonomous operation of HERMES, ensuring it can function independently in various scenarios.

2 Design

The Centre for Robotics and Future Production (CRFP) at the CSIR typically follows a reduced and fast-tracked V-Model development cycle [3]. The “V” of the V-model is divided into two sides, where the descending left side represents decomposition and definition of the problem, while the ascending right side of the model is the process of integration and verification of the solution. The system design starts with the analysis of the user needs and development of the system requirements. The system requirements then feed into the sub-system design for the mechanical, electrical and software sub-systems.

2.1 Requirements

The development cycle starts with the translation of a set of user needs to a feasible set of system requirements. These are then progressively decomposed into requirements for successive lower-level elements. The integration and verification process then involves the verification of the lower-level elements against their requirements and successive integration and verification of higher levels up to the full system level.

For this application, the user need is for a low-cost autonomous ground vehicle to perform security surveillance and patrols over large outdoor areas. A list of overall requirements can be listed as follows:

- The system needs to be a low-cost patrol vehicle able to withstand the elements as it patrols an outdoor environment during both day and night.
- The operational environment for the robot is a typical environment where existing security patrols would operate, such as tarred or dirt roads. The robot platform must therefore be able to operate on a typical dirt road.
- The surveillance robot needs an ingress protection (IP) rating of IP54 rated to withstand normal weather conditions that the vehicle will endure during its inspections. These include dust and splashing water.
- The surveillance robot requires the ability to detect people and vehicles and notify a control room operator of the detections.
- The system requires a communication system that is capable of streaming video footage from the vehicle’s cameras to a control room for monitoring.
- On a single charge, it is required that the vehicle should operate for about 8 hours, the length of a typical shift, and the vehicle should travel at speeds between 5 and 10 km/h for reliable patrolling.
- The system is required to be able to autonomously follow global positioning system (GPS) waypoints while performing obstacle detection and avoidance.

2.2 Mechanical design

The mechanical design approach of the surveillance robot aimed to develop a sound mechanical and structural system while prioritising simplicity in assembly and manufacturing. To reduce development time and cost, the system design focusses on using off-the-shelf components wherever possible. The “Sparky 1060 W Electric Kids Off-road Quad Bike” chassis [4], was modified to allow electronic actuation of the brakes, throttle, steering and to get wheel speed feedback. The quad bike seat and body were removed and replaced with mounts for the electronic boxes and the mounting post for the surveillance cameras as shown in Fig. 1.

Fig. 1. The HERMES autonomous surveillance robot.

By following this approach, the design and manufacturing time were significantly reduced, leveraging pre-existing subassemblies such as the chassis, and suspension. With a key focus on autonomy, an analysis of the quad bike was conducted to identify essential mechanical modifications necessary to realise an autonomous system. The frame, steering, and braking systems were identified as the sub-assemblies that required modification.

2.2.1 Braking System

The quad bike's braking system utilises a conventional hand lever, cable, and brake calliper system. After studying the standard braking system, a force balance was performed by measuring the force required to brake, using a spring balance, and applying a static force analysis. This exercise aimed to establish the minimum actuator specification necessary to execute braking operations effectively. Taking the sum of moments about the pivot point 'O' in Fig**.** 2, the braking force on the actuator cable was calculated to be at least 370 N.

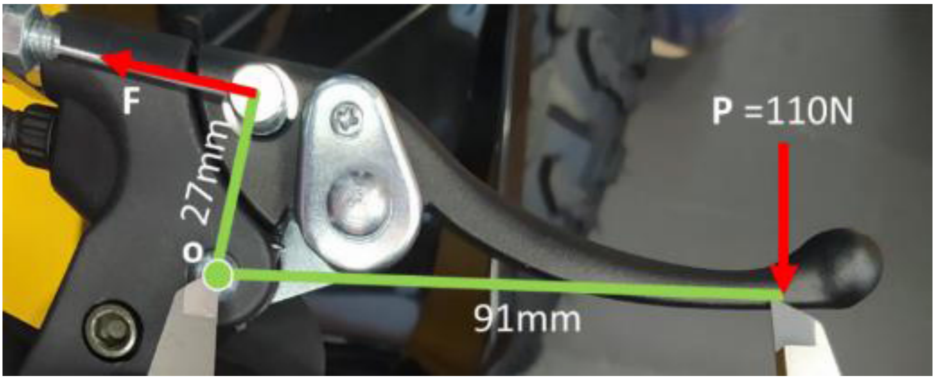

Fig. 2. Braking force free body diagram.

To allow for the autonomous actuation of the braking system, three concepts were produced to replace the hand-operated lever system with an electromechanical actuator. From the three concepts shown in Fig. 3, concept 2 was selected for its simplicity and ease of manufacturing. This concept works by directly tensioning the brake cables using an off-the-shelf 100 mm, 500 N linear actuator.

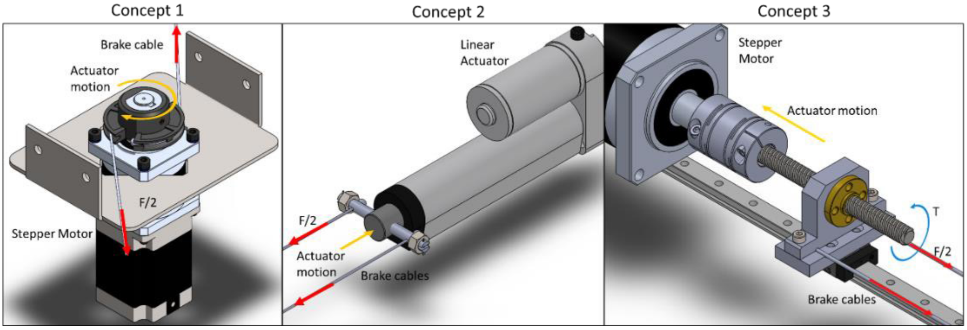

Fig. 3. Braking system concepts.

The linear actuator solely featured built-in top and bottom end limit switches, and to provide variable braking control, the actuator position feedback is required. This is achieved through the incorporation of a linear potentiometer which determines the position of the linear actuator, as shown in Figure 4. The mountings for the potentiometer and linear actuator were designed using computer aided design (CAD). The potentiometer mountings were produced using a 3D printed process implemented in (Polylactic Acid) PLA material, whilst the mountings of the linear actuator were fabricated from steel using a water jet cutter, and these were then welded to the frame.

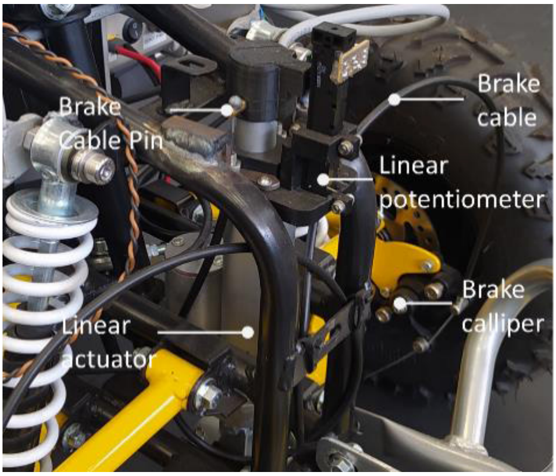

Fig. 4. Implemented braking system with feedback.

2.2.2 Steering System

The main challenge in achieving actuated steering was generating the required torque. To address this, the original steering handle was replaced with a 3:1 gearbox and stepper motor configuration. The gearbox case was fabricated using a waterjet cutter and standard off-the-shelf gears were assembled into the case. Similar to the integration of the linear actuator in the braking system, steering position feedback is required for the control system. Hence, an absolute rotary encoder was mounted on the driven gear end. Figure 5 depicts the implementation of the steering system.

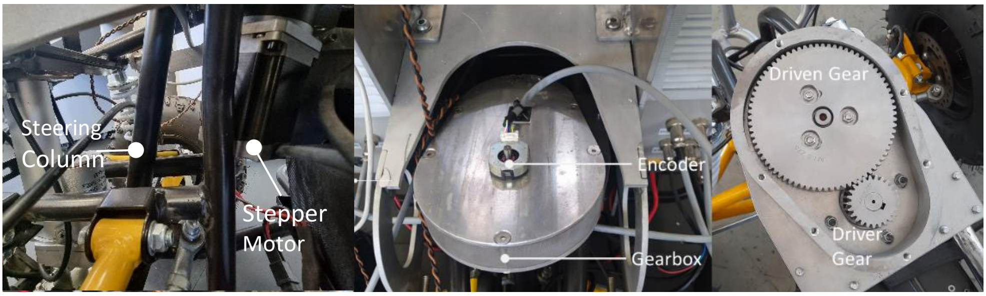

Fig. 5. Surveillance robot steering system

2.2.3 Structural Design

The primary objective of the structural design is to provide rigid structural support for the electrical components, actuators, and sensors. The mounting structure replaces the quad's original seat and body work with a frame fabricated using waterjet cut 5 mm and 8 mm aluminium sheet. The mounting structure provides mounting areas on each side for the two electronics boxes, a front mount for the 3D scanning LiDAR and a central mount for the camera mounting pole which provides an elevated position for the surveillance cameras.

2.3 Electrical design

The overall electrical system design, shown in Fig. 6, can be divided into three primary sub-systems: the quad control sub-system, the main processing and sensing sub-system, and the surveillance camera sub-system. In addition to the main sub-systems there are also the supporting power supply and communications components.

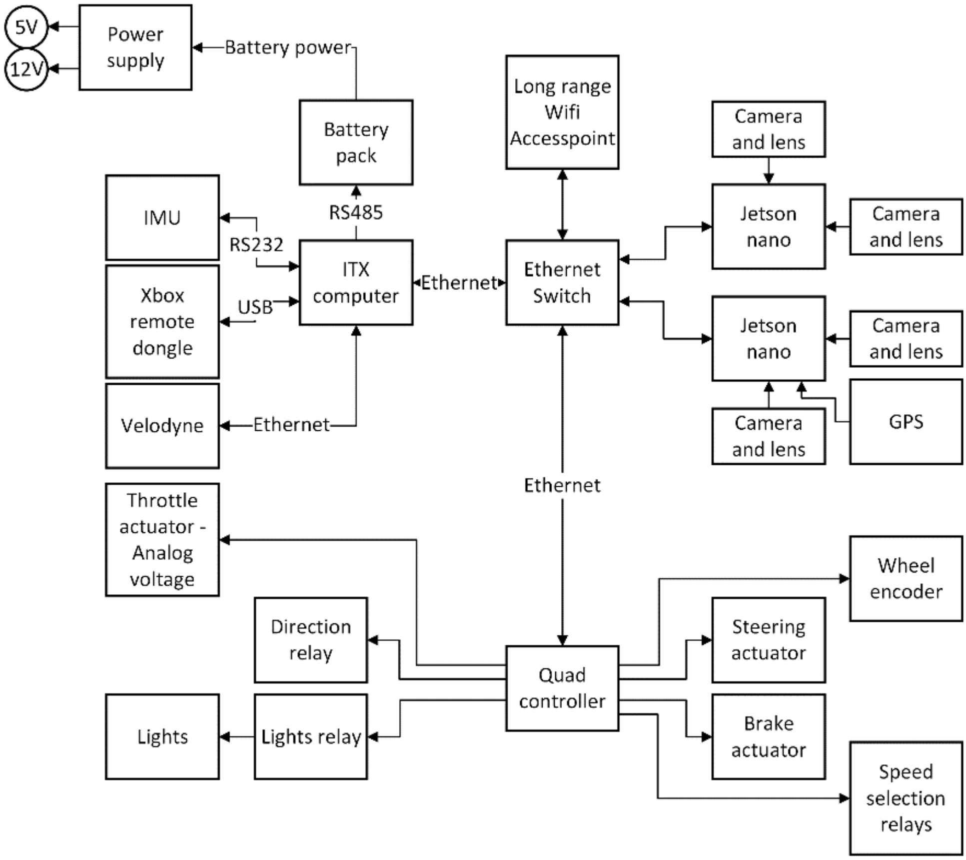

Fig. 6. High-level electrical system design for the surveillance robot

The quad control sub-system is responsible for the low-level interfacing with and control of the quad bike itself. The quad control sub-system consists of the quad controller, a Raspberry Pi Pico microcontroller connected to an Ethernet-to-UART converter and controllers for the various actuators and sensors for feedback. The electric quad bike is controlled through an analogue throttle voltage signal as well as speed and direction relay signals. These signals directly replace the throttle and switches on the original quad bike allowing for the use of the quad bike's controller without modification. An inductive proximity sensor reading a slotted disk on the rear axle provides speed feedback to the controller. The quad controller outputs signals to a stepper motor controller for the steering and a DC motor controller for the brake actuator.

The main processing and sensing sub-system is responsible for collating the sensor information and sending commands to the quad control system to provide the platform autonomy. At the heart of the sub-system is an industrial i9 ITX computer. The computer is directly connected to the Velodyne HDL32E multibeam LiDAR, an MTi 630 Inertial

Measurement Unit (IMU) and a wireless game controller for remote control of the platform. The computer will also interface with a battery management system over RS485 but this is not yet integrated with the system. The computer also connects to the quad controller, long-range Wi-Fi and surveillance sub-systems through a Gigabit Ethernet switch.

The surveillance camera sub-system consists of two Nvidia Jetson Nanos, four IMX477 camera modules with wide angle lenses and a Real-Time Kinematic (RTK) GPS receiver module. Each Jetson Nano streams video from two cameras. Additionally, both Jetson Nanos run on-board object detection on one of their streams, and one of the Jetsons receives GPS fix information.

2.4 Software design

The software architecture for the surveillance robot follows the CRFP's standard architecture for autonomous ground vehicles, as shown in Figure 7. The architecture is based around ROS2 [5] and the Nav2 stack [6]. The Nav2 stack performs the global path-planning and local control and obstacle avoidance using the robot description and robot specific configurations tuned for the specific robot use case and kinematics.

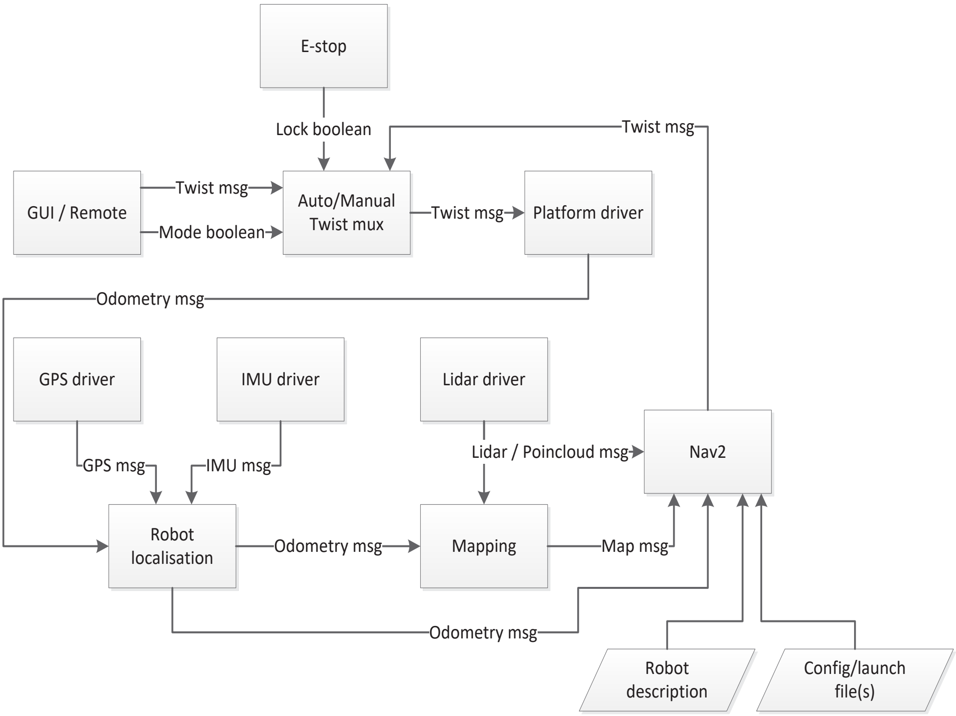

Fig. 7. A high-level autonomous ground vehicle software overview

The *slam_toolbox* package [7] is used to perform simultaneous localization and mapping (SLAM) and provides a global map for the Nav2 stack to perform global path planning. To build the map, this toolbox makes use of the odometry, and the sensor information received from the Lidar scans. The toolbox provides two mapping modes of operation: synchronous and asynchronous mapping. We opted for online asynchronous mapping, as it provides for real-time processing of measurements, as opposed to using pre-recorded data. In asynchronous mode, new measurements are only processed when the last measurement is completed, and the update rate criteria is always met ensuring that the system never lags in real-time. This is especially important when running complex loop closures.

The robot localisation package runs an extended Kalman filter-based sensor fusion to provide improved odometry estimates to the navigation stack. The robot localisation runs two filters: one for the local odometry and one for the global odometry. The local odometry fuses the odometry estimate from the platform driver with IMU data. The global odometry fusion uses a *navsat_transform_node* to transform GPS fix data into the robot's map frame and fuses this GPS odometry with the IMU and platform odometry.

The platform driver consists of two nodes working together to control the motion of the platform. The platform control node is a basic TCP server interface for communicating with the quad controller (see Fig. 6) over the Ethernet network. This node sends speed and steering angle commands to the quad controller and receives feedback on the current speed, angle, throttle and brake percentages. The embedded software on the quad controller is responsible for control of the brake and throttle to track the setpoint that comes from the platform control node. The second node is responsible for taking a standard ROS2 Twist message and calculating the speed and steering angle commands required. In addition, this node uses the feedback from the platform control node to calculate and publish the robot odometry used for localisation and navigation.

The *twist_mux* package multiplexes the velocity commands from the remote and from the Nav2 autonomy. The commands from the remote are given a higher priority than those from the autonomy allowing the user to take over manual control at any point. The twist mux allows all commands to be stopped using a lock Boolean, allowing the vehicle to be stopped from various stop signals.

2.4.1 Surveillance system

The surveillance system runs as an independent module on the Jetsons Nanos in the surveillance camera box. The sensing system uses the DNN Inference Nodes for ROS/ROS2 [8], specifically the DetectNet node for object detection. The detection node uses the SSD-Mobilenet-v2 model [9] trained on the MS COCO [10] dataset to detect object and publishes the detection information as well as an overlay image showing the object bounding box. The Jetsons are connected to the rest of the system through the Ethernet network making the detections available to the rest of the network. The detections of people and vehicles can be streamed remotely over the Wi-Fi link as well as being stored on the vehicle itself together with time and GPS metadata for later analysis.

3 Results

The testing of the system involves the verification and integration activities of the right-hand side of the V-model vee. This involves working up from the lowest sub-system levels and verifying that that sub-system meets its requirements and can be integrated into the next system level. This process continues up to the final system level.

The requirements relating to the mechanical sub-system are the ability to drive on typical tarred and dirt roads at speeds of between 5 km/h and 10 km/h, integrate the sensors and actuators required for autonomous control of the platform and provide a mounting base for the electrical system. All this needs to be achieved with an environmental protection rating of at least IP54. The base quad bike platform is designed to drive on the types of terrain that the surveillance robot is intended for and driving tests conducted on the CSIR campus confirmed its abilities. Testing of the platform throttle response shows that the quad is easily capable of driving at the desired 5-10 km/h. The throttle response curve (see Fig. 8) shows that the quad is capable of a top speed of 10 m/s (36 km/h) while retaining good low speed control because of its multiple speed selection settings.

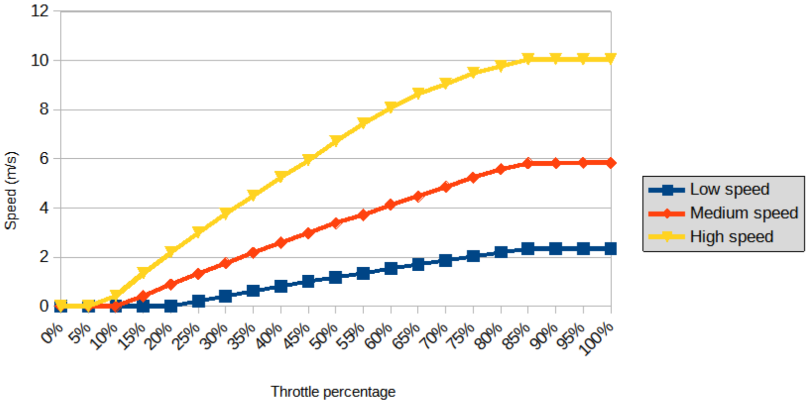

Fig. 8. Speed versus throttle percentage graph for each speed setting.

The steering and brake modifications provided the required electromechanical actuation to effectively control the platform from a computerised system allowing autonomous and remote drive functionality. The surveillance robot currently does not meet the IP rating requirement however this will be addressed with the addition of protective body work that will provide environmental protection and improve the visual appearance of the system. The design of the body is in progress and will be added to the system at a later stage without significantly affecting the integration of the other sub-systems.

The electrical sub-system provides the power and interfacing to the sensors and actuators required for the system autonomy. The electrical sub-system provides all the interfaces, allowing communication with the surveillance robot and its automated and manual navigation.

The original lead-acid batteries powering the quad bike were replaced with a large 30 Ah, 12 cell lithium iron phosphate battery to extend the operating time while performing surveillance. Figure 10 shows the current and battery capacity usage for HERMES driving a 1.5 km route over 26 minutes. From Figure 9 and the total battery capacity the full run time while continuously driving and performing surveillance and detection is 6.5 hours with a patrol route length of up to 22.5 km. While a 6.5 hour run time is nominally below the required 8 hour run time it is expected that HERMES would stop from time to time during a typical patrol which would extend the run time.

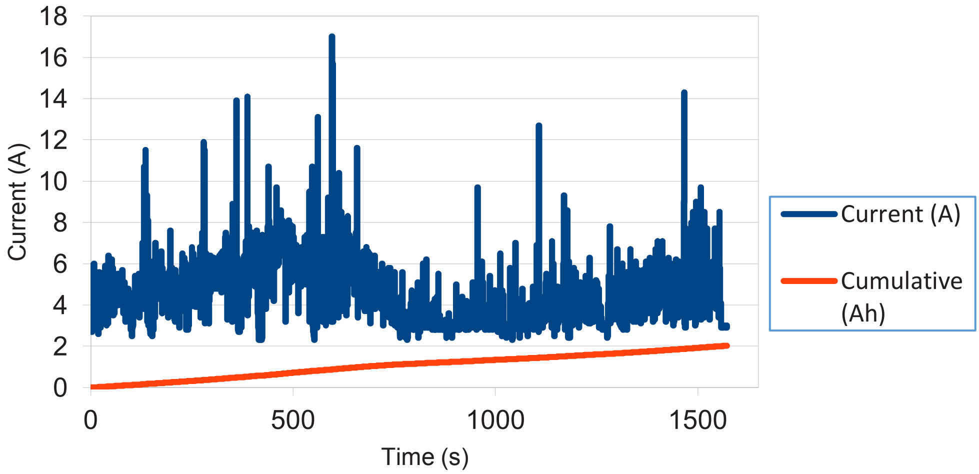

Fig. 9. Battery current and cumulative capacity used by HERMES driving at typical walking speed.

For the communication sub-system, a Software Defined Radio (SDR) based mesh network system was initially tested. The radios did provide a good line-of-sight range but had limited bandwidth and was very prone to obstructions to the line-of-sight. At the maximum bandwidth setting the SDR radios could only stream a single 2048 kbit/s video stream. Any obstruction to the line-of-sight between the radios immediately cut the stream. This could be resolved by using additional radios in the mesh however this was not considered feasible given the number of nodes that would be required in a typical urban environment and the extremely high cost of the SDR radios. It was therefore decided that a Wi-Fi mesh network provided a better alternative. The range of the Wi-Fi nodes are significantly less than the SDR ones, but they provide substantially higher bandwidth and are a fraction of the cost, allowing for a denser mesh.

The surveillance sub-system successfully runs person and vehicle detection at a rate of 3 frames per second (FPS) on two video streams while being able to simultaneously stream the four raw video streams at 7 FPS. All the processing for the object detection is performed on-board the surveillance robot. The frame rates may not provide the smoothest video feeds but are usable and close to the 10-15 FPS commonly used in the video surveillance industry [11]. The quantitative performance of the object detection, with outputs shown in Figure 10, was not evaluated since a standard detection model trained on a standard dataset was used and it is expected that the performance would match the performance for the model. Qualitatively, the detection performed acceptably with occasional misclassifications which are not expected to negatively affect the usability of the system.

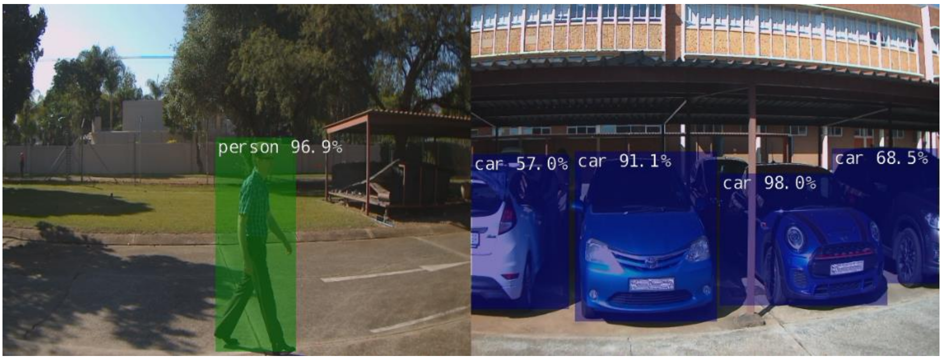

Fig. 10. Overlay images showing detections of a person and cars.

The autonomous navigation using the Nav2 stack was first tested in Gazebo [12] simulation before deployment and testing on the physical platform. Testing on the physical platform is still in progress. Currently, the platform has demonstrated the ability to autonomously navigate to a defined waypoint however the navigation is not smooth. The robot follows a snaking path towards the goal as shown in Fig. 11. This behaviour was not seen in simulation and after extensive testing of the localisation and navigation software it is believed that the problem lies in the path following controller.

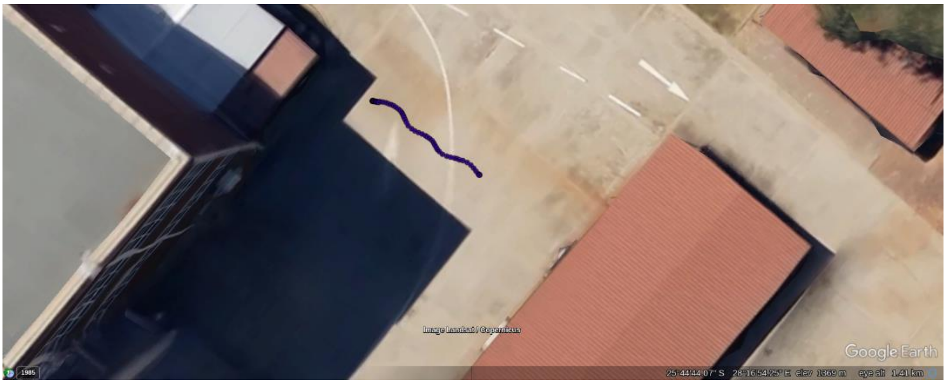

Fig. 11. Snaking autonomous navigation path

4 Conclusion

A low-cost surveillance robot platform has been designed for autonomous security patrol applications. The design approach focused on the use of off-the-shelf components as far as possible with the robot platform being based on a modified small electric quad bike. The surveillance robot can currently perform real-time person and vehicle detection, video streaming, manual and limited autonomous navigation on a low-cost platform.

The surveillance robot currently fulfils most of the key requirements for the system with additions and improvements in development. The speed, coverage area, and object detection ability of HERMES during surveillance tasks has been assessed. The speed of the robot is more than sufficient for the expected surveillance task. The robot can patrol a route of up to 22.5 km while running object detection.

The cost of HERMES has been kept low using an off-the-shelf electric quad bike base with simple modifications using primarily off-the-shelf mechanical and electronic components.

The modifications have equipped the robot with the required sensors and actuators to enable autonomous navigation however further work is still required to improve the autonomous navigation.

The design of the quad body is still ongoing with future work including the design of body panels to address the insufficient IP rating of the platform and additional testing and tuning of the autonomous navigation. Further future work will include the integration and verification of a large-capacity battery pack and the manufacture and integration of the quad's body work. Future work will also include testing of HERMES in a more realistic test deployment.

References

1. World Population Review, "Crime Rate by Country 2023," 2023. [Online]. Available: https://worldpopulationreview.com/country-rankings/crime-rate-by-country.
2. Transnet SOC Ltd, "Cable Theft Stats," 2023. [Online]. Available: https://www.transnet.net/Media/Pages/Cable-Theft-Stats.aspx.
3. K. Forsberg and H. Mooz, "The Relationship of System Engineering to the Project Cycle," in *The 12th Internet World Congress on Project Management*, Oslo, 1994.
4. "Sparky 1060w Electric Kids Off-road Quad Bike – Yellow," SK Imports, 2021. [Online]. Available: https://gokartsquadsbikes.co.za/product/quad-bikes/sparky-electric-offroad-quad-yellow/ . [Accessed 28 June 2023].
5. S. Macenski, T. Foote, B. Gerkey, C. Lalancette and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics,* vol. 7, no. 66, p. eabm6074, 2022.
6. S. Macenski, F. Martin, R. White and J. G. Clavero, "The Marathon 2: A Navigation System," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, 2020.
7. S. Macenski and I. Jambrecic, "SLAM Toolbox: SLAM for the dynamic world," *Journal of Open Source Software,* vol. 6, no. 61, p. 2783, 2021.

8. D. Franklin, “DNN Inference Nodes for ROS/ROS2,” 2023. [Online]. Available: https://github.com/dusty-nv/ros_deep_learning.

9. Y.-C. Chiu, C.-Y. Tsai, M.-D. Ruan, G.-Y. Shen and T.-T. Lee, “Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems,” in *International Conference on System Science and Engineering (ICSSE)*, Kagawa, Japan, 2020.

10. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll'ar and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision--ECCV 2014: 13th European Conference*, Zurich, 2014.

11. IPVM , “Frame Rate Guide for Video Surveillance,” 18 January 2021. [Online]. Available: https://ipvm.com/reports/frame-rate-surveillance-guide. [Accessed 24 May 2023].

12. N. Koenig and A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, 2004.