Design and Implementation of an Autonomous Mobile Robot for Slug Detection and Safe Collection to Prevent Agricultural Damage

Nabil Miri [1], Jana Marzouk [1], Abdulrahim El Mohamad [1], Yahya Aljamal [1], and Hassan Hariri [1,2,]*

[1] Department of Mechanical and Mechatronics Engineering, Rafik Hariri University, Mechref, Lebanon
[2] ECAM LaSalle, Lyon, France
Email: MiriNH@students.rhu.edu.lb (N.M.); MarzoukJB@students.rhu.edu.lb (J.M.); ElMohamedAA@students.rhu.edu.lb (A.E.M.); AljamalYY@students.rhu.edu.lb (Y.A)
*Corresponding author

***Abstract*—This paper introduces an innovative robotic solution to address the challenge of slug damage in agriculture. Slugs, particularly the grey field slug, represent a global plant pest that poses a significant threat to various crops. Their capacity to damage crops not only affects the quality of the product but also leads to unsellable vegetables in supermarkets, making it a pressing concern for gardeners and farmers. Existing methods for slug control often involve labor-intensive hand-picking or the use of chemicals, which can have detrimental effects on the environment and human health. This research presents an eco-friendly and smart solution that ensures the well-being of slugs while effectively addressing this agricultural challenge. The Robot Operating System (ROS)-based autonomous mobile robot, equipped with a camera and employing the YOLOv5 (You Only Look Once) model for slug detection, autonomously navigates agricultural environments using Global Positioning System technology, ensuring precise localization. The collection mechanism, thoughtfully designed to capture slugs without harming them. In contrast to other slug-killing robots, this solution focuses on the safety of slugs, making it a slug-friendly approach. The collected slugs are then safely deposited in a designated storage area. The resulting proof-of-concept robot is both functional and cost-effective, offering potential for scalable production.**

***Keywords*—slugs, autonomous navigation, computer vision, You Only Look Once (YOLO), Robot Operating System (ROS)**

I. Introduction

In modern society, advanced technologies are continuously being integrated to enhance everyday life and address unforeseen challenges. The robotics industry, in particular, has made significant progress in high-level decision-making, even in complex environments [1], demonstrating its value across industries seeking efficient solutions [2]. One such industry is agriculture, which has experienced notable advancements in integrating robotics and Artificial Intelligence (AI) due to its potential for enhancing agricultural productivity and promoting economic growth [3]. The monitored climate change has an impact on the reproduction rate of various species, including slugs such as grey field slugs, etc. which are widely considered as global plant pests. These creatures are among the most destructive garden pests and pose a significant challenge to gardeners and farmers due to their tendency to damage a broad range of crops and living plants, such as potatoes, tomatoes, cabbage, lettuce, strawberries, melons, bulbs, and ornamental flowers [4].

The destruction of plants due to their cosmetic appearance is a prevalent issue, potentially resulting in vegetables deemed unsellable in supermarkets. The consumption of plant roots by slugs poses a significant threat to garden development, hindering seedlings from attaching themselves to the ground and receiving essential nutrients [4].

The surge in the population of slugs, which pose a threat to local plants in households, has led to the adoption of multiple control measures. One of the simplest methods is hand-picking, where gardeners search for slugs on a daily or weekly basis, focusing on their hiding areas, which are usually watered in the afternoon and searched after dark. This method is manual, tiring and often inefficient. Chemical solutions, such as Metaldehyde and Mesurol, raise concerns, prompting many organizations and experts to advocate and encourage eco-friendly approaches such as traps, utilizing halved fruits like melons or beer-filled jars aimed at controlling slug infestations [5]. According to Gonzalez-de-Santos *et al.* [6], robotic systems have shown potential as effective solutions for pest-related problems in agriculture, leading to increased crop quality and improved health and safety for production operators. The SlugBot is a collaborative initiative involving SRC, AI experts COSMONiO, and farming enterprise AV and N Lee, seeking to address the challenge of slug-induced crop damage. The proposed solution centers around a specialized spray designed robot to eliminate slugs, with a primary focus on large-scale farms [7]. In contrast, our solution is tailored specifically for backyard farms.

Manuscript received February 15, 2024; revised March 14, 2024; accepted April 15, 2024; published July 19, 2024.

Another slug-killing robot, the MSR-BOT-PROJECT, emerged in 2016 and was developed at the University of Kassel. This robot utilizes digital image processing to identify slugs and employs nails to neutralize them effectively [8].

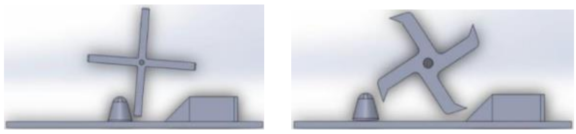
The third project is developed by the University of West England in 2001 where microbial fuel cells were used to convert slugs into biomass and generate electricity. The SlugBot9000, which is based on this technology, uses a vision sensor and a 360-degree extending arm to locate and grab slugs and then drop them into an onboard trap [9]. Recently, a project was done by [10] where they developed a state-of-the-art optical filter-based system to detect slugs. They measured the visible and Visible Near-Infrared (VNIR) waveband of the slug and the soil, and found that the slug is 12.4 times brighter. Thus, differentiating between slugs and soil. The paper only discussed the feasibility of this method but didn't have any robotic prototype. Unlike previous solutions, some lacking prototypes or relying on slug extermination, and others designed for expansive farms, our proposed robotic solution stands out for its slug-friendly approach. It ensures the safe disposal of slugs without causing harm. Moreover, our robot is purposefully tailored for home backyards, characterized by smaller dimensions and cost efficiency. Our autonomous robot is equipped with a camera employing YOLOv5 for slug detection and navigates autonomously using ROS and GPS system. Moreover, the drivetrain is designed based on the backyard's rough environment. Detected slugs are collected safely through a well-designed mechanism and stored in a specific container.

II. Robot Details and Implementation

The design of the robot involved the development of four main parts, namely Collection, Drivetrain, Detection, and Navigation. These parts were designed independently and then integrated together to form the complete robotic system.

A. *Collection*

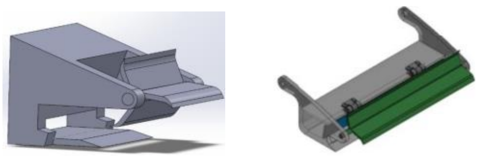
Picking a small and slimy creature such as a slug necessitates the use of a specialized collection mechanism that can effectively capture the slug without it sticking to the mechanism. Prior to the development of the final mechanism, several concepts were considered and evaluated based on various criteria, including the number of motors required, the level of precision needed, the depth factor obtained from the RGBD camera, cost, and the potential harm to the slug. One proposed mechanism was a robotic arm with a soft gripper, which could grasp the slug and deposit it into a drawer. While effective and precise, this approach was deemed unsuitable due to its high cost and complexity. Another idea involved a suction tube with a gripper, which could vertically capture the slug and transport it to a chamber within the robot. However, the added complexity, cost, and precision required for positioning rendered this approach unfeasible. A bulldozer-like mechanism was also considered, which would be lowered to collect the slug without necessitating high precision. However, it was found to be problematic due to its tendency to collect dust and push the slug instead of collecting it. Consequently, a ramp-like mechanism with a roller was chosen as the optimal solution. When the robot detects the slug, the ramp descends to ground level, and the roller begins turning as the robot approaches the slug. This method effectively collects the slug within the ramp using the roller. The final design of the Roller mechanism underwent several iterations before being selected. This mechanism is composed of three main components, namely, a roller to push the slug inwards, a ramp that facilitates smooth sliding of the slug towards the storage, and a storage unit where the collected slugs are kept prior to disposal. In the design process, two different rollers were initially tested, one with sharp edges and the other with square edges, as depicted in Fig. 1.



(a) (b)

Fig. 1. CAD of Roller with (a) Square Edges and (b) Sharp Edges of the Collection Mechanism.

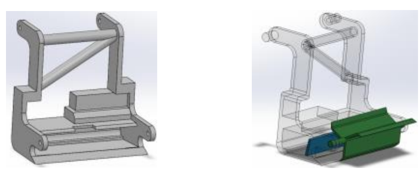
To test the efficacy of the Roller mechanism, both versions of the roller with sharp and square edges were fabricated through 3D printing and tested on real slugs. The experiment is elaborated in Section III.A. The design has undergone several changes. The first iteration was a basic roller with a ramp, which proved to be ineffective. Subsequently, in the next iteration, the mechanism was enlarged, and a storage compartment was added along with a blue one-way gate to prevent the slugs from going out again as shown in Fig. 2(b) and Fig. 3(b).



(a) (b)

Fig. 2. Design of the (a) first and (b) second iterations of the ramp.

In the final iteration, the mechanism was reduced in size and support was added, and the integration points were improved with even reduced space between them.



(a) (b)

Fig. 3. Design of the final iteration of the collection mecahnism (a) without the roller and (b) with the roller.

B. *Drivetrain*

Designing the drivetrain depends on the testing environment, robot speed and weight, and the collection mechanism. For the robot to navigate through rough agricultural terrain, two possible options were found: continuous tracks or four wheels. While tracks have the benefits of less ground impact and the potential for lower weight growth, they may also have lower speed, less maneuverability, and be more difficult to repair. Alternatively, the use of four wheels has the advantages of lower cost, higher speed, high maneuverability, lightweight design, and simplicity. But driving over obstacles may present a challenge. After weighing all the advantages and disadvantages of the available options, we have decided to design a robot with a four-wheeled drivetrain.

Aluminum profiles were utilized for the robot chassis, for their lightweight, robustness, and durability. These profiles are available in standard sizes, with the 2020 (20×20 cm thickness) being the most ideal option. However, since the T-nuts required to connect the profiles were unavailable in the local market, we choose the 3030 profiles. We based our decision on weight considerations, having performed estimations for the anticipated weights of the robot chassis using both 3030 and 4040 profiles as shown in Table I.

TABLE I. CHASSIS' WEIGHT (IN KILLOGRAMS) OF DIFFERENT ALUMINUM PROFILES [LENGTHS IN METERS]

Profile Size	Upper Part	Lower Part	Middle Layer	Total Length	Mass per unit length [Kg/m]	Total Weight
2020	0.940	0.850	0.280	2.070	0.48	0.9936
3030	0.900	0.810	0.280	1.990	0.9	1.791
4040	0.860	0.770	0.280	1.910	1.46	2.7886

Selection of the motor is based on the speed and torque formulas below:

$$\text{Speed[RPM]} = 60 \times \frac{\text{NominalSpeed}}{(\pi \times \text{wheelRadius} \times 2)} \quad (1)$$

$$\text{Torque[Nm]} = \frac{\text{wheelRadius} \times \text{trustForce} \times \text{SafetyFactor}}{4 \times \text{efficiency}} \quad (2)$$

where:

$$\text{trustForce[N]} = \text{gravity} \times \text{frictCoe} \times \text{robWeight} \quad (3)$$

The robot weight was estimated to be 6.023 kg, taking into account the components to be used and a safety factor of 1.5. The nominal robot velocity was set to 0.75 m/s, which was deemed sufficient as speed was not a primary factor. A nominal robot acceleration of 0.25 m/s^2 was also specified. The drive wheel diameter was 0.125 m. The friction coefficient between the tire and mud was calculated to be approximately 0.158 [11]. A motor efficiency of 75% was assumed, and a safety factor of 2.5 was included in the calculations. Using the above equations and params, we got: Torque required is 3.59 kg.cm and 114.6 RPM (Rotation Per Minute) Speed. Our drivetrain was designed on Solidworks before being implemented. The use of CAD/CAE technology allows designers to create 3D models of robots that can be used to simulate and test robot performance under different conditions. Thus, optimizing robot designs to improve performance, reduce costs, and enhance safety [12]. Fig. 4 illustrates the final CAD model of the robot. This model provided the precise dimensions required for the robot. As a result, the assembly was too smooth without any main designing issues.

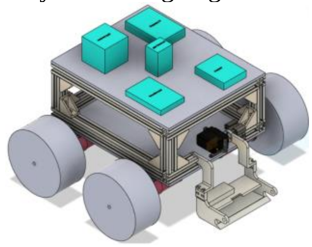


Fig. 4. Side-front view of the robot CAD's design.

The two main power sources in our robot are a lithium battery and a power bank. We calculated our battery sizing based on two modes which are illustrated in Tables II and III. The first method relies on 50% of the maximum current each component could consume during operation, ensuring robust performance under worst-case scenarios.

TABLE II. METHOD 1: BATTERY SIZING CALCULATION FOR WORST-CASE SCENARIO (50% OF MAXIMUM CURRENT)

Component	Current in A	Explanation
4 DC Motors	4	50% of Stall Current (2×4×50%)
Servo Motor (Lifting)	0.4	50% of Current at Max Load (0.8×50%)
DC Motor (Roller)	0.097	Continuous Current (A): 97mA
4 Motor Driver	0.06	
Capacity	3.4178	Ah: Sum of the Currents × 45 min (1 Cycle Operational Time)

TABLE III. POWER-BANK SIZING

Component	Current in A	Explanation
Raspberry Pi	2	
Arduino Mega	0.0732	
Capacity	1.5549	Ah: Sum of the Currents × 45 min (1 Cycle Operational Time)

In the second method as in Table IV, consumption of components is calculated using its operational time percentage:

TABLE IV. METHOD 2: BATTERY SIZING BASED ON THE OPERATIONAL TIME OF COMPONENT CURRENTS

Component	Current in A	Explanation
4 DC Motors	3.6	Operates for 90% of the time
Servo Motor (Lifting)	0.04	Operates for 10% of the time
DC Motor (Roller)	0.034	Operates for 35% of the time
4 Motor Driver	0.06	
Capacity	2.8005	Ah: Sum of the Currents × 45 min (1 Cycle Operational Time)

C. Detection

The robot was equipped with vision to enhance the slug's recognition and grasping, and raspberry pi RGB camera v2 was used due to its low cost and local market availability. To integrate detection in the robot, we built and deployed a YOLOv5 Model. First, we collected 150 images of slugs in different conditions such as lighting, surface type etc. Only one online dataset was found, but all the images in the available dataset are close-up images, which is not the case for our robot as the camera will be placed inclined on the robot and will have a wider range of view. Then the collected images were annotated. The annotation is done by drawing a boundary box (bbx) around the slugs. The data was split between 134 training and 16 testing images. YOLOv5 model was trained for 270 epochs and on 416 image size.

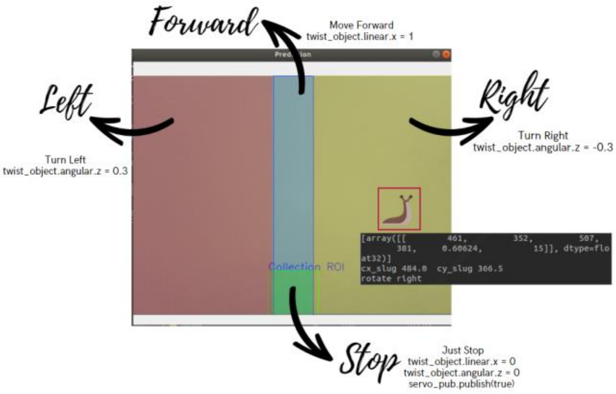


Fig. 5. Region of interest method.

Because of the unavailability of depth info, extracting the estimated position of the slug in 3D space using only the RGB camera was done using Region of Interest (RoI). The image window was divided into 4 parts, namely Forward, Right, Left, and Stop as demonstrated in Fig. 5. The robot motion follows the previous actions based on the slug's location in these regions. For example, if the slug was detected in the right region, then the robot will move right until the slug's position becomes in the Forward region. Then, the robot moves Forward until the slug is in the Stop region. Reaching this region gives the signal to the robot to drop the collection mechanism and collect the slug. Moreover, the slug's coordinate position (x, y) in the image is the center of the bounding box (red point in Fig. 11(c)) calculated from the detection model output.

D. Navigation

1) Navigation approach

To ensure the autonomy of the agricultural robot, we used advanced features of the Robot Operating System (ROS) along with multiple sensors [13]. ROS is a set of software libraries and tools that help to build robotic application [14]. The navigation system involved mapping, localization, path planning, and obstacle avoidance.

2) Mapping

In the case of the agricultural robot, the team used GIMP software to draw the map of the working environment, including all the static obstacles. The digital map, which is known as occupancy grid, consists of pixels that represent physical distance according to a specified resolution.

3) Localization

The Marvelmind GPS System is a high-precision localization solution for autonomous robots. It utilizes mobile beacons attached to the robot and stationary beacons interconnected through ultrasonic transducers and a wireless radio interface. The system provides real-time position updates with an impressive ± 2 cm precision. A modem is used for wireless communication between components, which can be connected to a PC for configuration, data optimization, and visualization [15]. Notably, the system automatically creates a map of stationary beacons without requiring manual data input or distance measurements, making it user-friendly and convenient for robot localization.

The robot's rotation is determined using an MPU-6050 sensor module, which includes a 6-axis motion tracking device with a gyroscope and accelerometer.

4) Obstacle avoidance

After generating an occupancy grid for the navigation environment and determining the location of the robot within the grid, two additional maps are created: the global costmap and the local costmap. The global costmap remains the same size as the main map of the environment, and it is static. On the other hand, the local costmap is a dynamic map that moves with the robot, and it is determined by the size of the robot. Various sensors can be utilized for obstacle detection, ranging from IR sensors to cameras. Once the obstacles are detected and put inside the local costmap, the path planner, later, will take them into consideration and commands the car to avoid these obstacles in real-time. In our case no sensors were used for avoiding dynamic obstacles as our testing field had only static obstacles that are already defined in the Global Costmap and the planner avoids them successfully. The parameters of costmaps, such as global frame and robot base frame etc., were edited using their Yaml files (a configuration file).

5) Path planning

In our case, the user sends the initial goal to the robot through RViz (RViz is a ROS graphical interface that allows the user to visualize information such as robot position, transformations, markers etc., and using plugins, such as global and local cost maps, for many kinds of available topics), and the robot uses the DWA (Dynamic Window Approach) planner to choose the best path to reach the goal.

III. Experimental Results, and Discussion

All the previously discussed parts are integrated together to end up with our robot. The main steps were to integrate the collection mechanism designed with the drivetrain and combining the vision model with the navigation to guide the robot to the slug location. This part shows the experiments and results of the collection mechanism design, battery sizing and the vision metrics.

A. Collection Mechanism Experiments

A quantitative test was done to conclude the best collection speed or roller motor rotational speed. This was based on slug size, orientation of the slug (Orientation 1: slug parallel to the roller blade and Orientation 2: slug perpendicular to the roller blade). A total of 48 trails were conducted, 24 for each speed, which concluded that using the 60 RPM speed is more efficient. The experiments were done on a flat surface. The test data is shown in Table V.

TABLE V. ROLLER MOTOR SPEED TESTS

Speed	Slug Size	Orientation	Successful trials	Percentage
35 RPM	M	1	6	79.16%
		2	6	
	S	1	3	
		2	4	
60 RPM	M	1	6	95.83%
		2	5	
	S	1	5	
		2	5	

B. Components Selection: (Motors, Motor Driver and Battery)

Motors's needed specifications availability in the Lebanese market were very limited. Thus, the motors chosen were a bit oversized. After searching all available options, we ended up with 2 choices: (1) 100 rpm, torque 12 kg.cm, stall current 2 A and (2) 120 rpm, torque 18 kg.cm, stall current 7A. We traded off some needed speed to get a motor with much less stall or max current to avoid buying a motor drive with a bigger max current rating of 7 A. Thus, due to power consumption reasons, 2 A motors were chosen. The motor serial number is 37GB520 and the model is 17320-70. The motor will operate at a torque of 4kg.cm and the motor's speed will be 66.67 RPM according to motor specifications.

The motor driver should have a maximum current rating equals or more than the motor's stall current. The selected motors have a stall current of 2 A; therefore, the double bridge L298N motor driver is chosen since it can hold a maximum of 2 A drive current in each bridge [16]. Two motor drivers were needed for the four motors required for the robot. Based on the calculations in Section II.B and the market availability, we chose a 12 V 4400 mAh 18650 Lithium-ion Battery which is higher than the worst-case scenario (3.4 Ah).

C. Vision Metrics

The mean Average Precision (mAP) is a commonly used evaluation metric for object detection models, which measures the accuracy of the model in detecting and localizing objects in an image. The mAP50 measures the average precision when the overlap between the predicted bounding boxes and the ground truth bbx is at least 50%. A high value of 97% for mAP50 indicates that the model performs well in terms of accurately localizing and recognizing objects in the images. On the other hand, mAP50-95 represents the average precision calculated when the overlap threshold ranges from 50% to 95%.

According to Table VI, the mAP50-95 was 58%, less than mAP50 (97%). The main reason is the low number of images, but overall, the YOLOv5 model was sufficient for our proof of concept. Moreover, Fig. 6 illustrates the Precision-Recall Curve of the trained model.

TABLE VI. YOLOV5 MODEL METRICS

Dataset	Precision	Recall	mAP50	mAP50-95
Training	0.938	0.969	0.976	0.576
Validation	0.936	0.969	0.978	0.579

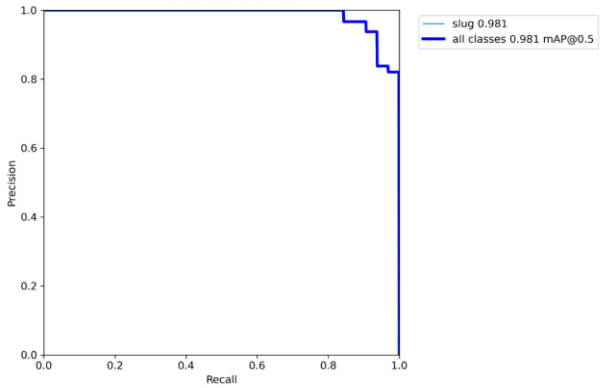


Fig. 6. Precision-recall curve.

D. Drivetrain, Collection Mechanism and Integration

The robot chassis, as shown in Fig. 7, is built using 3030 aluminum profiles, connected by L-type brackets and M5-T nuts, washers, and screws. Two wooden plates with a thickness of 7 mm were cut down and fixed to the bottom and top of the chassis with M5 T-nuts, washers, and screws. The wheels used for the robot are φ12.5 cm in diameter as per specifications and were attached to the motors using φ6mm motor couplings.

The implementation of the collection mechanism design was done using 3D printing technology (using PLA or Polylactic acid material). The mechanism was found to operate effectively, with consideration given to environmental constraints. Specifically, the terrain where the slug is located must be flat to ensure proper operation of the mechanism. The motor responsible for the roller rotation is a normal geared 12 v DC motor with maximum speed of 60 rpm. A simple motor bracket is used to mount the motor on the collection mechanism.

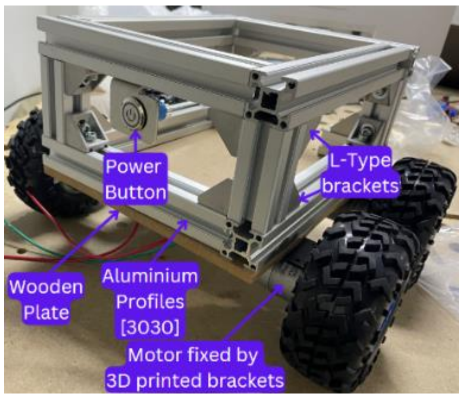
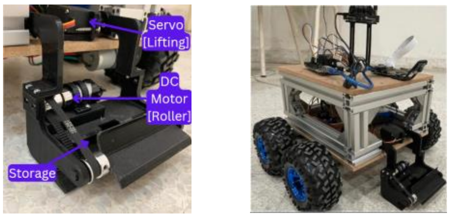


Fig. 7. Robot chassis with motors and wheels.

As shown in Fig. 8(a), a gear and a helping belt are used to transfer the rotational movement of the motor to the collection roller. Another MG995 Servo is responsible for the radial lifting motion of the mechanism. The mechanism is attached to the front of the drivetrain and the space in between is left for the servo motor that's responsible for lifting the mechanism in angular fashion making the navigation of the robot doable.



(a) (b)

Fig. 8. Final (a) Collection mechanism and (b) Robot prototype.

E. Vision Part with the Navigation

The integration of the velocity commands from navigation and detection is a crucial aspect of robot motion control. As shown in Fig. 9, the Twist multiplexer package from ROS is utilized to combine the velocity commands from planner and vision [17]. The twist_mux package offers a multiplexer for geometry_msgs: Twist messages, which selects the messages from a single input topic based on priority and timeout.

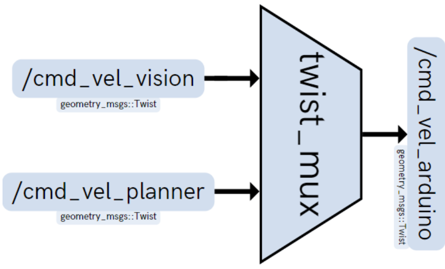


Fig. 9. Twist_MUX.

In the present scenario, the priority is assigned to the velocity command from vision, and the robot navigates according to this command whenever the camera detects a slug, else the robot will follow the navigation velocity input. We edited the Yaml files of the twist_mux topics where we assigned priority of 1 for the cmd_vel_planner and priority of 2 for cmd_vel_vision [2 has higher priority than 1]. In Fig. 10, the robot navigation information and communication schematic is shown.

At the end, several experiments have been done on the final integration of the robot in a constrained environment to test the robot and solve some problems that may occur. Constraints included flat surface for collection mechanism, operation done in daylight, and less obstacles in the surrounding. Thus, it was done in a classroom.

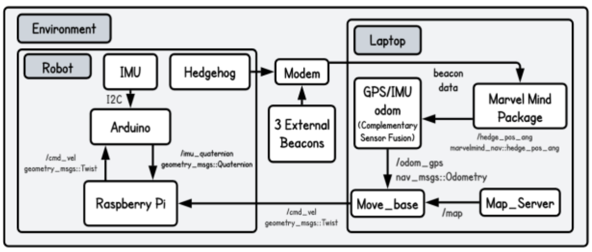


Fig. 10. Robot navigation schematic.

A map of the environment was given beforehand to the robot. A slug was placed inside the environment and the robot was given the points it should go through as shown in Fig. 11(a), where the red line is the potential path to follow and inside the green circle is the slug. While the robot is autonomously navigating, it continuously stays checking for a slug and when it finds one, the robot temporarily ignores the navigation velocity commands and goes towards the slug. Using RoI method discussed in Section II.C or Fig. 5, and when the slug is in the right position for collection, as shown in Fig. 11(c), the robot stops, and descend the collection mechanism, Fig.11(d) The collection mechanism motor starts rotating and with the robot's forward motion, the slug is successfully collected. The demo of the robot can be found in the reference. A demo of the robot can be found in [18].

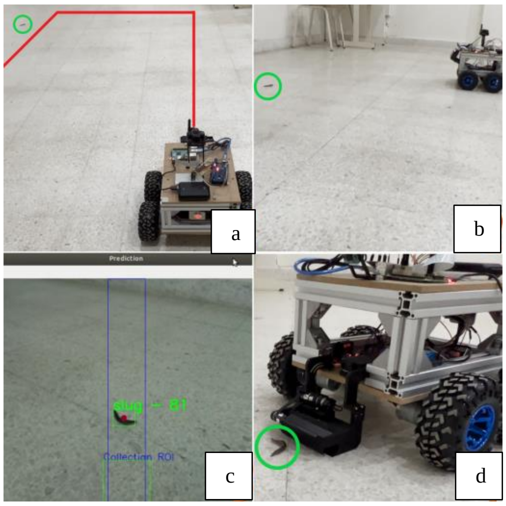


Fig. 11. Slug collection procedure with robot camera feed image.

IV. Conclusion And Future Work

In this paper, we presented the systematic integration of all the system hardware and software aspects that resulted in a functioning prototype that proofs the concept. The robot can navigate autonomously, detect slugs (achieving mAP50 of 97%, and mAP50-95 of 58%) and collect them safely with a success rate of 95.83%, taking into account some pre-defined constraints. The current prototype serves as a foundational step towards a better robotic system in subsequent iterations. Notably, testing was conducted in a controlled, flat terrain, and occasional challenges were observed in the robot's differentiation between slugs and smaller objects.

Some of the future developments would include enhancing vision system through RGBD camera for improved depth perception and utilizing a GPU-equipped for decentralized processing. Navigation improvements involve transitioning to an outdoor GPS kit and incorporating obstacle avoidance. In the collection phase, the mechanism will be refined for rough terrains, adapted for slug-friendliness using flexible materials, and future upgrades will implement a one-way door to prevent slugs from escaping the collection mechanism.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Research, Y.A.; hardware, A.E.; software, N.M. and J.M.; supervision H.H. All authors wrote the paper, revised the manuscript and approved the final version.

Funding

This research was funded by Spexal.

References

[1] J. Wirtz *et al.*, "Brave new world: Service robots," *Journal of Service Management*, 2018.
[2] J.-C. Liao *et al.*, "Designing and manufacturing of automatic robotic lawn mower," *MDPI Processes*, 2021.
[3] V. Marinoudi *et al.*, "Robotics and labour in agriculture. A context consideration," *Biosystems Engineering*, vol. 184, pp. 111–121, 2019.
[4] W. Golder. (2023). Are slugs good for gardens?—An expert's answer. *Florida Yards*. [Online]. Available: https://www.floridayards.org/are-slugs-good-for-gardens-an-experts-answer/
[5] Slugs Facts and Info. [Online]. Available: https://www.rhs.org.uk/biodiversity/slugs
[6] P. Gonzalez-de-Santos *et al.*, "Fleets of robots for environmentally-safe pest control in agriculture," *Precis. Agric.*, pp. 574–614, 2017.
[7] CHAP. SlugBot aims to put an end to slimy pests. [Online]. Available: https://chap-solutions.co.uk/news/slugbot-aims-to-put-an-end-to-slimy-pests/
[8] J. Gödeke *et al.*, "Development of a robotic solution to detect and fight slugs," in *Proc. 11th Young Scientists Meeting 2018*, Braunschweig, Germany, 2018.
[9] I. Kelly *et al.*, "SlugBot: A robotic predator in the natural world," in *Proc. AROB: 5th International Symposium on Artificial Life and Robotics*, 2001.
[10] C. Höing *et al.*, "Development of an optical system based on spectral imaging used for a slug control robot," *Horticulturae*, vol. 8, no. 1, 77, 2022.
[11] G. Somareddy. (2018). [Online]. Available: https://www.quora.com/What-is-the-coefficient-of-friction-between-mud-and-a-tyre
[12] M. Chugunov and I. Polunina, "Interdisciplinary modelling of robots using CAD/CAE technology," *Mordovia University Bulletin*, vol. 28, no. 2, pp. 181–190, 2018.
[13] ROS-Robot Operating System. [Online]. Available: https://www.ros.org/
[14] N. Shalal, T. Low, C. McCarthy, and N. Hancock, "A review of autonomous navigation systems in agricultural environments," in *Proc. 2013 Society for Engineering in Agriculture Conference*, 2013.
[15] J. Huang *et al.*, "Indoor positioning systems of mobile robots: A review," *MDPI Robotics*, 2023.
[16] Interface L298N DC Motor Driver Module with Arduino. [Online]. Available: https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/
[17] E. Fernandez and S. Gevatter. (2015). twist_mux. [Online]. Available: http://wiki.ros.org/twist_mux
[18] Slugs Picking Robot. [Online]. Available: https://www.youtube.com/watch?v=bieaTB9a4QQ

Copyright © 2024 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.