Long Range Drone System
Drone Team 1

Theodor Paal Põlluste thpul23@student.sdu.dk
Benjamin Thordahl Christensen bench19@student.sdu.dk
Lucas Rønne Jeppesen ljepp19@student.sdu.dk
Thomas van der Sterren thvan23@student.sdu.dk

***Abstract*—Current development within uncrewed aerial vehicles largely focuses on developing robust aerial systems capable of performing many different autonomous sensing tasks. However, a significant limiting factor for all these sensing tasks is the maximum area a drone can cover. One way autonomy is added to this principle is by using drone-in-a-box principles, though this still limits flexibility for range. Within this study, a novel approach is investigated to increase this flexibility while maintaining autonomous capabilities. This study explores a UAV-UGV hybrid platform, enabling extended-range operations through docking-capable UGVs and UWB-based localization. To achieve this symbiosis, a reliable near-field localization system is investigated using Ultra Wideband sensors, ROS2 communication, and hardware development to allow the agents to perform their tasks together. The results prove that, although some parts require more attention, the possibility of such a platform is not only realistic but also usable.**

***Index Terms*—UAV, Drones, Observation, Surveying, UWB, Ultra Wideband**

I. Description of the industrial challenge

The use of Uncrewed aerial vehicles for surveying and monitoring is growing. International initiatives are contributing to research in this field, such as the European Union's Horizon 2020 program [1]. One of the biggest bottlenecks during this research remains the flight time of UAVs and thus their limited range. For this reason, some companies are investing in alternative power methods beyond the industry's common Lithium Polymer Batteries. With the main goal being, more and longer autonomous flights. An example of such a platform is the Skyfront Perimeter 8, a hybrid gas-electric drone offering extended flight endurance [2]. Addressing this issue of long flight is crucial to accelerating progress in aerial monitoring. This report discusses an alternative method to the one previously mentioned.

This report will review current research related to this project, present a novel approach, and demonstrate a possible implementation of a long-range system. The report will conclude by reflecting on the feasibility of this design for monitoring and surveying purposes.

II. State of the art

A. Aerial Vehicle Monitoring

As stated earlier, there are multiple ways to approach this issue. In addition to alternative energy sources, a common solution is hybrid UAV systems. Platforms such as the CW-30E VTOL UAV developed by JOUAV demonstrate this approach. VTOL, which stands for Vertical Takeoff and Landing, combines the vertical takeoff and landing capabilities of a quadcopter with the range and endurance of a fixed-wing drone [3]. Equipped with advanced sensors such as high-resolution cameras and LiDAR, these UAVs are widely used for mapping, surveillance, and inspections. However, despite these improvements, most UAVs still rely on Lithium Polymer (LiPo) batteries, which limit their operational range and require frequent recharging [4]. To overcome these constraints, integrating UAVs with UGVs presents a promising solution for extending range and mission duration, particularly in applications requiring long-distance autonomous monitoring.

B. Ground Vehicles

Ground travel, whether crewed or Uncrewed, have been around for a long time. In 2004, DARPA, the US Defense Research Organization, posted a challenge for autonomous off-road locomotion. [5] After this challenge multiple companies have already emerged with solutions, both civil and military. A few examples would be Capra Robotics [6], Catapilar [7], and Milrem Robotics [8]. The range of these systems is much more reliable, additionally, it is less of an issue for these systems to have additional payload systems, in our case a UAV and its required equipment.

C. Poor Satellite Navigation Environments

The access to good Global Navigation Satalite System (GNSS) coverage is not a given. Especially when out in rural areas where tree coverage, weather and terrain can create unpredictable circumstances, it is difficult to rely a precision system, for a task like landing on a platform, to satalite systems such as GPS. For this reason, alternative systems must be explored. Landing on vehicles is nothing new either. In 2019, a team worked on landing a drone on a moving ground platform using ROS. With which they succesfully did so. [9]. Later in mid 2024, two researches built untop of this, using an Ultra Wideband localization to achieve an estimation of relative location, after which a Apriltag, a form of Aruco marker, would then be used to do precise landing. Thus still remaining in need of a visual landing system. [10]

D. Drone-in-a-box

Currently, drone-in-a-box systems sold commercially are sold for surveillance [11] or site evaluations [12]. In research, there are not that many articles written about this subject alone. It is mainly used as a tool for the main focus of the research. In medical use cases, such as delivering needs between hospitals, the largest company is Zipline [13]. The main approach on the market is to have a small lightweight drone that has a pyramid-like shape having a centering system built into the shape of the drone. Mainly the drone-in-a-box solutions are stationary thus there is no need to lock the drone into the box. For charging mainly pogo-pins are used, having gravity apply the contact force between the UAV and the box. To achieve higher ranges a different UAV type is used. Using a VTOL(Vertical take-off and landing) [14] type drone with wings. Meaning it uses lift generated from passive wings instead of propellers pushing the UAV off the ground.

III. Approach

It is well known that ground vehicles have a much larger operating range, additionally, they consume little to no energy remaining stationary. Combining these platforms to create a multi-agent system would allow long-distance autonomous travel, between areas of interest. On top of that, it would make it possible to survey different areas in one mission. In figure 1, an illustration is portrayed of how such a system would be implemented. As can be seen, the Uncrewed Ground Vehicle (UGV) will be tasked to travel to, and in between, different areas of observations. The Uncrewed Aerial Vehicle (UAV) will fly from the UGV Landing/take-off pad and will conduct its aerial monitoring. In the meantime, the UGV can move to the end of the UAV's flight zone, to allow it to land. This way the UAV's airtime is minimized, allowing it to save energy for more flights. After landing the UGV will travel on to the next location of interest.

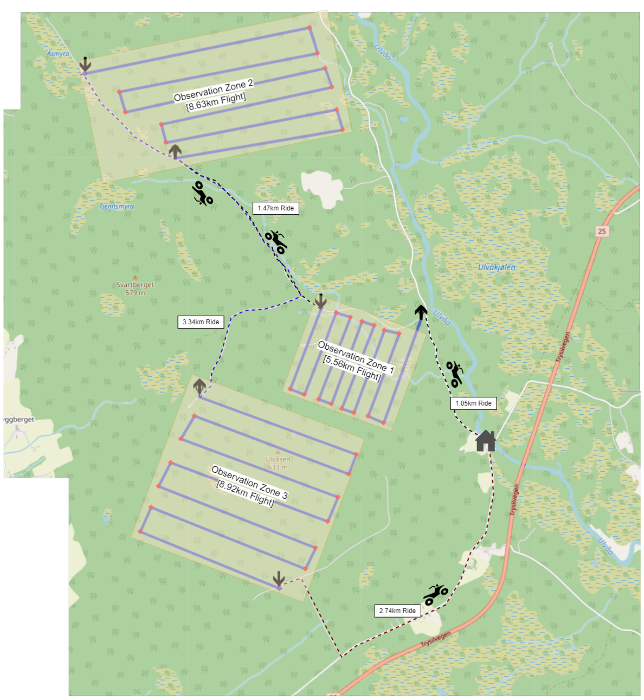

Fig. 1. A Surveying mission laid out as an example.

To achieve this goal, a couple of different tasks must be completed and challenges must be overcome. It is key to create an experimental prototype capable of demonstrating the approach to get a sense of reliability. The main challenge that will be addressed within this prototype will be the use of a multi-agent platform. Specifically, the creation of a UGV, and its landing pad with a locking mechanism, The autonomous lift-off, motion, and landing for the UAV in ROS2 [15], and finally the safe landing procedure in GPS unstable environments using alternative methods, being Ultra Wideband.

IV. Experimental prototype

The UGV is built on an all-terrain vehicle, remounted to allow remote operation. The UGV was reverse-engineered to be used with current software and communication.

A. *Uncrewed Ground Vehicle*

The UGV is equipped with a DC motor with an encoder controlled by a BAC-0501 motor controller [16]. Turning the wheelbase was done using a linear actuator. On top of the UGV a landing pad was built, see section IV-B, with a locking mechanism to secure the drone when landing, which used Dynamixel [17] to open and close. The main computer of the UGV is a Raspberry Pi 3B+ [18], from now on referred to as Rasberry. The Raspberry and the ground station uses Ubuntu 22.04 LTS, and ROS2 Humble for communication. The UGV control is done via an Xbox One controller connected to the ground station.

1) ROS enviroment: The ground station is running a ROS2 node that publishes a Twist message [19]. This message contains values for six degrees of freedom, three for linear values, and three for angular values. To drive forward and reverse the UGV the x-linear value is used and turning uses the z-angular. The ground station sends at 50 Hz to be close to real-time and responsive control. By setting Quality of service to **Besteffort** the message takes the current input of the controller instead of having a buffer with all inputs. Using a best-effort approach, communication may be lost, but it will not waste time trying to send old commands to the UGV. The UGV has two nodes, one listening to the Twist message with control commands and another listening to the landing pad command. The control command sends signals to the I/O port of the Raspberry, which is connected to the Motor controller. The values used for testing were greatly reduced to ensure the UGV did not run wild. The Landing Pad node initializes the Dynamixel motors, see section IV-B for details on the landing pad. The node sets up serial communication and functions for torque, opening and closing the motors. When an opening command is received it would open the locking mechanism and wait for a close command. The ROS2 network for the full system can be seen in figure 4

B. *Landing-pad*

The interface between the UGV and the UAV is a landing pad on top of the UGV. It consists of a simple plate with a centering and locking mechanism. Which currently consisted of a sliding scissor mechanism. This was done, since centering is required for potentially charging of the UAVs batteries and a more predictable take-off position. Due to the UAV's T-shaped legs, the other centering methods become cumbersome. The usual parallel rod mechanism [20] would require lengthy linear rails and belts. The team came up with a compact, mechanically simple, and robust system that is installed in the center of the landing pad. The system consists of an actuator that drives two pulleys with a cable transmission. The transmission allows the pulleys to rotate in opposite directions from the same input. The pulleys have rods installed tangentially, with the rods overlapping each other at the central plane of the landing pad, thus the name sliding scissor mechanism. This is true for each motor position except when the rods become parallel. That attribute is exploited to move the leg of the UAV in the central direction. A representation of the system used on the landing pad is visible in figure 2. The mirrored system

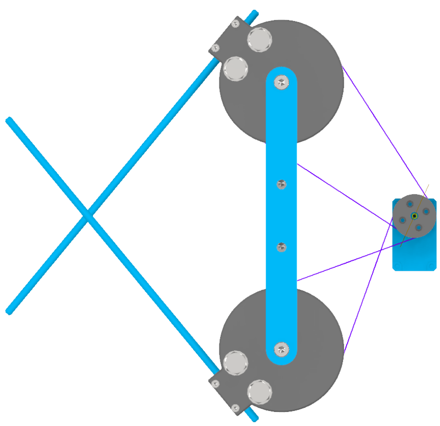

Fig. 2. Sliding scissor mechanism CAD preview.

ensures that the two mechanisms work against each other, moving the UAV's legs away from the centroid of the landing pad. The motor runs until a torque limit is reached. If the torque limit is not reached, it can be assumed that the landing has gone wrong and the sliding scissor mechanism returns to the unlocked position. The UAV can take off and land again. After a successful centering the motors are deactivated and the friction from the motor's gearing is enough to hold the position, saving energy. The motors can also produce holding torque if rougher terrain is expected. In the locked position the rods reach over the T-shaped legs clamping it down onto the landing pad keeping the UAV locked in all degrees of freedom.

1) Experimentation: To acquire proper data on the functionality of the landing and locking mechanism, more preciously the centering aspect, an experiment was set up.

a) Centering Robustness Experiment: The UGV and the UAV were placed inside a calibrated Motion Capture system to record the 3D positions accurately. The UAV was positioned in random positions and orientations on the landing pad. The sliding scissor mechanism was activated and the motors ran until the torque limit was reached. The 3D position of the UAV was registered and then repositioned randomly. The experiment consists of multiple runs to retrieve enough data. This data will later be reviewed in the Results section.

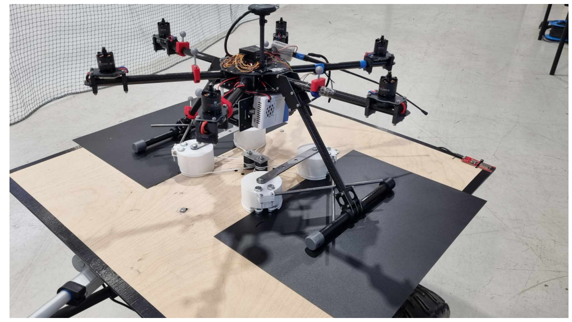

Fig. 3. Supplied UAV on developed landingpad.

C. Uncrewed Aerial Vehicle

1) Hardware Modification: The UAV that was used, shown in figure 3, was a Hexcopter with Pixhawk provided by SDU.

2) ROS Environment: ROS2 is used for drone navigation. The ROS nodes can be seen in figure 4 where the *GO-TO*, *Path*, *TakeOff*, *Landing*, *UWB*, and *MavROS Drone* nodes are running on the Raspberry Pi on the drone.

a) GO-TO: This Node facilitates drone control to avoid race conditions by handling and sending the set positions to the drone. Its action server uses interpolation between the drone's current and target positions to fly in a straight line and limit the drone's speed. Thus, it does not fly between two points directly. It uses the drone's current point or last target point as the set point to keep off-board mode available. As well as implementing a service to take control in cases where more manual control is needed, as seen in the Landing Node.

b) TakeOff: the TakeOff node implemented an action server for taking off to a specified height. The action server waits for the drone to be in offboard mode for an armed state either enabled by a ROS parameter or through manual arming. After this, a GO-TO action call will be sent with the target position begin the specific height above the drone's current position.

c) path: the Path node implements a service for sending the mission points as well as a latched publisher for the current running state. The service returns after receiving the point, confirming if the point is accepted or denied based on whether it is currently flying a set of points. After accepting a set of points, it will send an action call to GO-TO for each point in the received order. it was chosen to use a service instead of an action for the path node, as the drone will fly away from the mission controller and lose connection with the action client.

d) Landing: The landing node implemented an action server that uses the UWB to find its relative position to the UGV and 4 PID controller for x, y,z, and yaw velocity. Then, the action server is called, it takes control from the GO-TO node, the drone flies using velocity control on top of the UGV at the drone's current altitude with the yaw of the UGV for then to fly down and land on the UGV after which it changes the drone mode to landing and gives control back to the GO-TO node.

e) UWB: This node is currently designed to use an external truth in space for the Drone and UGV, to be used for calculating a relative position between them in the frame of the UGV, this is indeed to send out data as if the Ultra Wideband is used. Allowing for spoofing using a Motion Capture system

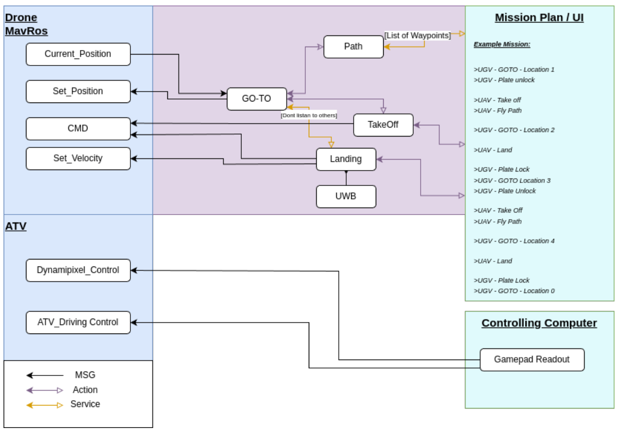

Fig. 4. ROS2 Node layout for Missions.

3) User Interface: A GUI(Graphical User Interface) was created to allow easy control in performing testing using the QT framework [21], as can be seen in Figure 5. The GUI gives the drone's local position as well as its ultra-wideband position, together with the drone's flight mode, and if it is armed and running any action at the moment, allowing for an easy overview of the current state. The GUI also includes a table of path points, where points can be added, removed, and changed before sending them off to the drone for flight.

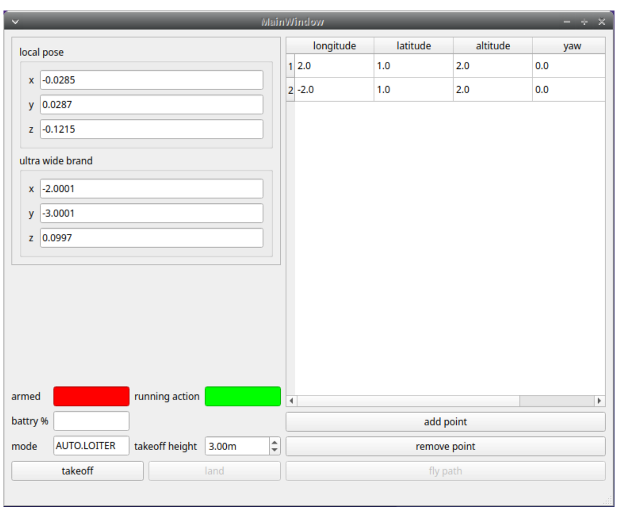

Fig. 5. Graphical User Interface

D. Ultra-Wide Band

For the UAV to land safely on the UGV, even in areas where the Global Navigation Satellite System (GNSS) might be low, or even restricted, an alternative approach was investigated. The use of a positioning tool called Ultra Wideband was researched. The theoretical benefits of such a system allowed the UAV to return to the landing pad without the need for visual aid, which could be obstructed by debris or dirt, or the need for fixed infrastructure for precise localization. This, in turn, would allow the system to be more widely usable.

1) Hardware for UWB: The UWB system tested is the Makerfabs ESP32 UWB development board, as seen in figure 6. These boards have the DW1000 chip-set onboard, capable of establishing an UWB communication with each other and pinging each other for range. Using the ESP32, allows developers to create easy scripts to interact with the chip-set. To get started with the development board, Makerfabs created a GitHub repository [22], allowing easy startup.

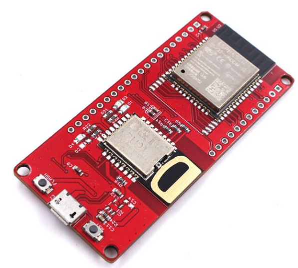

Fig. 6. Makerfabs ESP32 UWB development board.

2) Calculating Algorithm: To Gather three-dimensional data, an algorithm was implemented. The method used was by utilizing the 'scipy' library within Python. Since there are three distances to anchors measured, with each their offset (eg. anchor 1:$[x_1, y_1, z_1]$), it became possible to create three sets of equations:

$$\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2}-d_1=0,$$
$$\sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2}-d_2=0,$$
$$\sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2}-d_3=0.$$

The SciPy 'fsolve' function would return two possible answers, since it is known that the UAV is not underground, the assumption can be made to specify a single coordinate.

3) Experiments: To test the accuracy and implementation possibilities with the UWB system, a few experiments were conducted. To create a ground truth, a motion capture system was utilized. This system, at the time of experimentation, had an accuracy of sub-millimeter (0.86mm). All experiments have been performed through one session, creating a lower likelihood of skewed data due to environmental conditions.

The development boards were fitted with reflector trackers, so their location could be tracked within the motion capture system, as can be seen in figure 7. These markers can then be localized within the testing facility, shown in figure 8. The following tests were conducted in order to test the possibilities the UWB system provides:

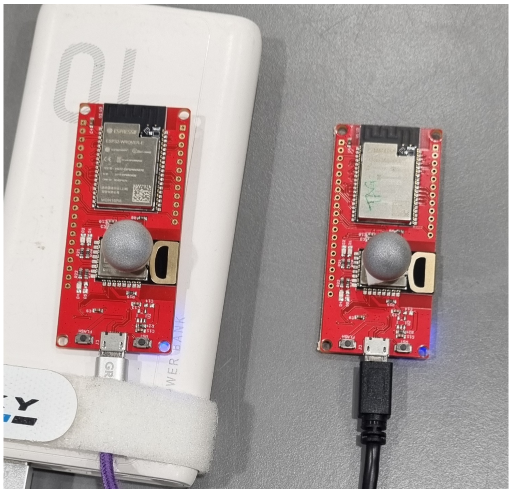

Fig. 7. UWB development board fitted with reflector trackers.

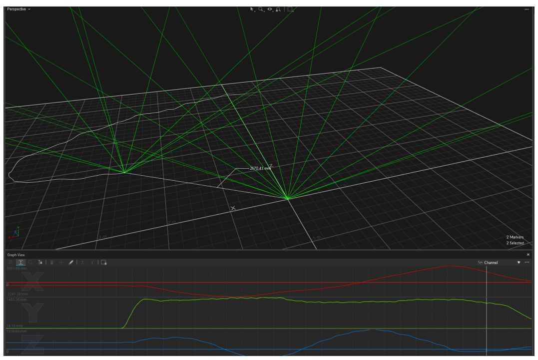

Fig. 8. Motion Capturing system localizing the trackers.

a) Single UWB Experiment: To test the distance accuracy, one anchor and the tag will be used. Where the distance is captured both in the Motion Capture environment, as well as through the tag. The tag will connect to a computer through Serial, and save its data in a CSV file for later analyses. The test will be conducted for a minimum of 2 minutes, to ensure sufficient data. Throughout the test, the tag and anchor distance will be changed (by means of moving through the motion capture environment).

b) Multi UWB Experiment: Similar to the initial test, however now, two more anchors will be introduced. To create usable data for the algorithm developed, the anchors will be placed at a known location. The locations of the anchors have been placed as follows:
$Anchor1 : (0, 0, 0)$
$Anchor2 : (-1000, 0, 0)$
$Anchor3 : (0, 1000, 0)$
These coordinates are then placed within the algorithm to create three-dimensional location data.

c) Localization Algorithm: Using the motion capture system, a "dummy" output can be generated, in the way the UWB would perceive the task. Thus, the data from the motion capture system would be used directly, as well as used to calculate distances between anchor and tag, these distances would then be written to a new CSV file, giving us the data that is aimed at. Overlaying this data with the actual location of each marker in time will indicate if the algorithm works.

d) Localization using UWB: Finally, the last experiment covers the data collected by the UWB modules themselves. The data will be fed into the algorithm giving an insight into whether the data is usable.

With all the data that is collected, it becomes possible to test the feasibility of the system as is.

V. Results

For the different experiments described in the above section, the results have been gathered. These have been divided among the different topics and displayed below.

A. Centering Robustness Experiment

The experiment ran 100 times and the X and Y coordinates were registered. The deviation from the center was calculated by subtracting the true center from the points registered. The data was tested for normality with the Shapiro-Wilk test ($p<0.05$, $W(200) = 0.89$). The results show that the data does not follow a normal distribution due to being heavily left-skewed. The main eigenvalue is at 167 degree angle to the X-axis showing that the skew is not only in one axis but there is more than one imbalance in the system.

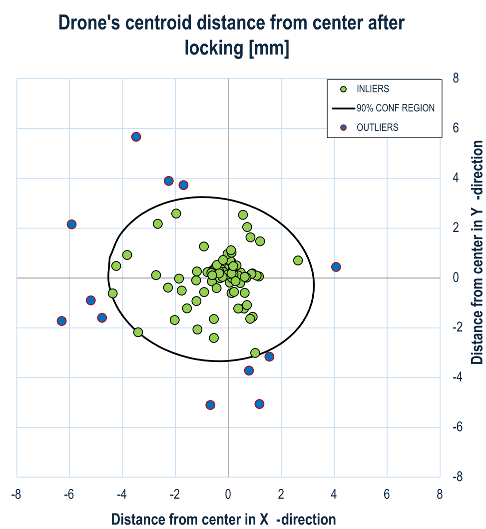

Fig. 9. Centering robustness experiment results

B. UWB Experiments

From the UWB experiments, a few interesting pieces of data have been collected. Looking at the graph in figure 10, it is possible to see that the UWB module follows the trajectory of the Motion Capture system well. When the UWB module stops moving it shows a slight offset to the motion capture system. This offset is within 10cm, which is within the specification of the module.

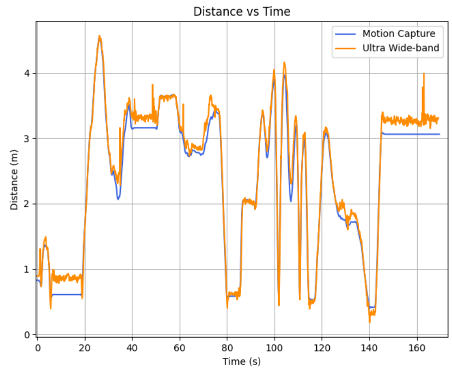

Fig. 10. Single Anchor and Tag distance result.

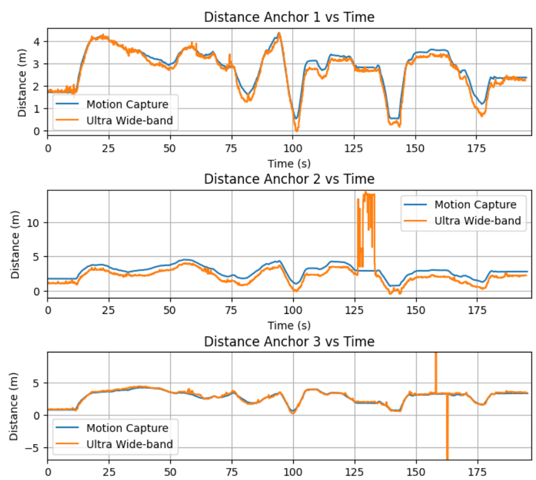

Fig. 11. Multi Anchor and Tag distance result.

In contrary to the previous test, the UWB module does not seem to hold constant in the multi-anchor test and is presented with a larger offset, as can be seen in figure 11. At some points measuring up to 40cm at certain points. Additionally, the distance to anchor 2 and anchor 3 had significant data fluctuations. To ensure a proper working algorithm, it is safe to test the setup with a known value. In this case, the position of the Motion Capture markers that are stuck to the UWB module. Overlaying the reformatted data, as described in the previous section, the data can be plotted. The results of which, shown in figure 12, indicate a functioning algorithm that has a few slight errors. For the most part of the path, the algorithm calculates the correct position based on the distances. These results allow for the testing with the actual collected data of the UWB module. Although similar results would have been expected, it is not what can be seen at all. In figure 13, the results of the same algorithm with the collected UWB data are displayed, where next to extreme outliers, no clear data is presented. Resembling nothing close to the expected results of a Motion-Capturing system.

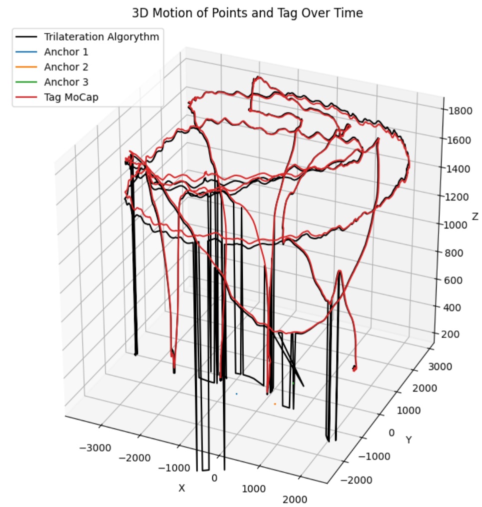

Fig. 12. Algorithm results using MoCap as serial in.

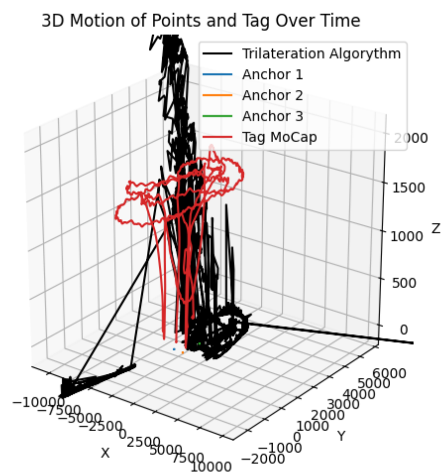

Fig. 13. Algorithm result using UWB serial in.

VI. Discussion

A. Centering Robustness Experiment

The results from the experiment show that the centering of the UAV works but has some downfalls. During testing, it was apparent that one side of the centering system works better than the other and the experiment shows that. The asymmetrical centering issue could be mitigated by connecting the outputs of the actuators. The data also shows that the pulley that moves in reverse to the motor's output has less pushing force or that the drone has more sliding friction moving in one direction than the other. One possible solution would be to use gears for transmission and change the contact points of the UAV's feet. To add a charging capability to this system the shortcomings need to be addressed to have a 99 percent confidence region have a smaller radius than 1 mm.

B. UWB Experiments

From the data collected, it became clear that using a single UWB module, the specifications of within 10cm were feasible, however, when multiple anchors were placed, the data seemed to become more inconsistent. Giving noise on the data, and additional outliers. A possible fault in the system could be antenna calibration, which was performed using a single module. This could increase the likelihood of the data being closer to the expected distance. Additionally, a filter could be investigated to ease the data that is collected, removing extreme noise and smoothing the motion.

The inconsistency of the collected data leads to the possible miscalculation of the algorithm resulting in an unusable system. The algorithm showed a working solution for the spoofed data, where some outliers and failures to calculate could be removed using a filter. As stated earlier, further investigation into the multi-anchor setup would be valuable for the implementation of this setup.

VII. Conclusion

This report investigated the development of an Uncrewed Ground vehicle platform that can be controlled with ROS2 communication. A landing pad that can center and lock the UAV in place was designed and implemented on the vehicle. Ultra Wideband modules were implemented on the UAV and UGV, and were used for positioning during autonomous landing. The aerial vehicle would use this system when executing the landing sequence. For managing missions, a User interface was created.

The centering mechanism performed reliably during tests but had minor inconsistencies, likely due to asymmetrical forces or mechanical imbalances. The mechanism requires further refinement to enhance precision and robustness, especially if it is to support integrated charging.

The experiments with Ultra-Wide-Band modules produced mixed results. Single-anchor configurations delivered accurate distance measurements within specifications. The multi-anchor setups had a lot of noise and occasional inaccuracies. Antenna calibration and noise-filtering methods could significantly improve performance in future iterations.

In each of these areas, valuable insight were gained and working prototypes was achieved. Further improvements must be made to produce a system that could be used in real-world problems.

References

[1] European Commission, "Horizon 2020: The eu framework programme for research and innovation." https://ec.europa.eu/programmes/horizon2020/, 2020. Available at https://ec.europa.eu/programmes/horizon2020/.

[2] Skyfront, "Skyfront perimeter 8: The longest endurance hybrid gas-electric multicopter drone." https://skyfront.com/, 2024. Available at https://skyfront.com/.

[3] JOUAV, "CW-30E: Vertical Take-off and Landing Fixed-Wing UAV," 2024. Accessed: 2024-12-17.

[4] N. Mehendale, "Investigating the battery life issues in unmanned aerial vehicles: An analysis of challenges and proposed solutions," *SSRN Electronic Journal*, 2021.

[5] J. Fulton and J. Pransky, "Darpa grand challenge – a pioneering event for autonomous robotic ground vehicles," *Industrial Robot*, vol. 31, no. 5, pp. 414–422, 2004.

[6] Capra, "Capra: Innovative Solutions and Technologies," 2024. Accessed: 2024-12-17.

[7] Caterpillar Inc., "Autonomy & automation," 2024.

[8] Milrem Robotics, "Milrem Robotics: Leading Robotics and Autonomous Systems," 2024. Accessed: 2024-12-17.

[9] P. R. Palafox, M. Garzón, J. Valente, J. J. Roldán, and A. Barrientos, "Robust visual-aided autonomous takeoff, tracking, and landing of a small uav on a moving landing platform for life-long operation," *Applied Sciences*, vol. 9, no. 13, p. 2661, 2019.

[10] J. Liu and C. Yan, "An integrated uwb-vision framework for autonomous approach and landing of uavs in gps-denied environments," 2024.

[11] V. Bene, Z. Kristóf, and B. Elek, "The dji dock 2 is a "drone in a box" to enhance the unmanned guard solution : Scientific research with the cooperation of duplitec ltd.," *Repüléstudományi Közlemények*, vol. 35, pp. 171–181, 01 2023.

[12] D. Enterprise, "Dji dock 2: Elevating automatic drone operations," 2024. Accessed: 2024-12-16.

[13] CNET, "New zipline drone brings aerial delivery closer for millions more of us," 2024. Accessed: 2024-12-16.

[14] JOUAV, "Jouav unattended system," 2024. Accessed: 2024-12-16.

[15] 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.

[16] Tech-Mobile, "BAC-0501 Wiring Diagram," 2024. Accessed: 2024-12-16.

[17] ROBOTIS, *XL430-W250 User Manual*, 2024. Accessed: 2024-12-17.

[18] Raspberry Pi Foundation, "Raspberry Pi 3 Model B+," 2024. Accessed: 2024-12-16.

[19] ROS Documentation, "geometry_msgs/Twist Message," Noetic. Accessed: 2024-12-17.

[20] igus®, "Drone in a box – igus® for aeronautical engineering," 2024. Accessed: 2024-06-16.

[21] "Qt — tools for each stage of software development lifecycle," 12 2024. accessed 2024-12-17.

[22] Makerfabs, "Makerfabs esp32 uwb github repository." https://github.com/Makerfabs/Makerfabs-ESP32-UWB, 2024. Accessed: 2024-12-12.

Acknowledgement

We would like to express our gratitude to Prof. Dr. Robert Ladig for his guidance and support throughout this project. His insights were instrumental in helping us combat the challenges within this field. We also extend our sincere thanks to the SDU UAS Center for providing access to their resources and testing facilities, which were critical for our research. Finally, we acknowledge the assistance of LLM AI in proofreading and debugging, which significantly improved the speed and quality of our work.

Individual contribution:

Theodor Paal Põlluste: Landingpad design, development, and testing. UWB functionality testing and ROS2 implementation.

Benjamin Thordahl Christensen: ATV Motion development and ROS2 implementation.

Lucas Rønne Jeppesen: UAV Flight path development and ROS2 implementation, UI Development.

Thomas van der Sterren: UWB ROS2-spoof development, UWB Development, and its Algorithm writing. UWB Testing. Project management.