

Mechatronic design of a delivery octarotor drone

Nikolaos Evangelou¹, Nikolaos Giakoumidis², Dimitris Chaikalis³, Athanasios Tsoukalas¹, Halil Utku Unlu³, Daitao Xing⁴, and Anthony Tzes¹*Senior Member, IEEE*

Abstract—This article describes the mechatronic design of a delivery octarotor drone equipped with safety features including protective bumpers for its propellers, parachute, and retractable gear. A GNSS-RTK unit provides extensive pose-refinement for trajectory tracking, while an RGB-Depth Camera provides distance measurements from any present obstacles. A deep learning algorithm classifies any objects into certain categories (i.e., balloons, drones) followed by a trajectory avoidance algorithm. An onboard computer handles these sensor measurements and communicates with the autopilot for adjusting the trajectory. At the same time, it provides HD-live video feed to the base station and runs any onboard ROS-services. The resulting drone can safely land within a circular area of radius 1.5 m, while flying over, under and around obstacles or buildings. The carried payload is handled by a dual linear actuator gripper that releases it upon the drone’s precise landing.

Index Terms—Delivery drone, Drone safety features, Obstacle classification

I. INTRODUCTION

DELIVERY drones made their experimental debut in 2014 for food and drink delivery, while similar efforts have been reported for postal services followed by recent commercial services for e-commerce and retailing. A recent survey on load transportation [1] shows an intensified research and commercial area. Hybrid schemes using drones and trucks [2] have been reported, while issues such as stationary recharging stations [3], [4]. Recent theoretical approaches have also appeared relying mostly on the TSP properly modified to account the drones’ peculiarities [5] while future advances in the area of drone delivery appear in [6], [7].

Ensuring safety to humans and property while operating these drones is of paramount importance [8], [9] and the governments are expected to regulate this market. Inhere, a commercial octarotor is used as a delivery drone with enhanced safety issues. The article describes these enhancements including the classical parachutes, propeller protective bumpers, alternative autopilots relying on local coordinates, GNSS-RTK for additional accuracy on the progressed path, RGB-D cameras for SLAM, obstacle avoidance while running deep learning algorithms for obstacle classifications as stationary (buildings) or moving (neighboring drones and balloons).

¹N. Evangelou, A. Tsoukalas and A. Tzes are with New York University Abu Dhabi, Electrical & Computer Engineering, UAE.

²N. Giakoumidis is with New York University Abu Dhabi, Core Technology Platform, UAE.

³D. Chaikalis and H. U. Unlu are with New York University, Electrical & Computer Engineering, USA.

⁴D. Xing is with New York University, Computer Science & Engineering, USA

II. OCTAROTOR DRONE DESCRIPTION

A. Octarotor design

The developed octarotor delivery drone relies on the frame of [Vulcan’s Mini 8](#). This has been equipped with a parachute, protective bumpers for the propellers, an RGB-D camera, GNSS-RTK for precision trajectory and a gripper with two parallel moving jaws for grasping and releasing the payload, as shown in Figure 1. The drone has a detachable gear can carry a payload of 3 Kgr (excluding its dual 16Ah 5S Lipo batteries). The flying envelope if this drone is over 15 mins with a 1.8 Kgr payload. The octarotor configuration offers increased robustness in flight against one or two motor failures.

Furthermore, the onboard depth sensors (x3 optional) have a Field of View (FoV) $87^\circ \times 58^\circ (\times 3)$ and measure the distance from 1280×720 points (obstacles) at distances 0.2 up to 10 m. At the same time, a deep learning algorithm operating on the HD-RGB video from this camera, identifies balloons and other drones and sends this information to the autopilot for obstacle avoidance. An onboard [Intel i7 NUC](#) running [Ubuntu 18.04](#), and the Robot Operating System[10] implements the Mavlink protocol and can assist the drone’s autopilot in case of emergency.



Fig. 1: Octarotor Delivery Drone

B. CUBE autopilot & on-board computer

The adopted Flight Control Unit (FCU) is the [CUBE](#) FCU, owing to its triple Inertial Measurement Unit (IMU) configuration for redundancy and open-source hardware implementation. On the software side, the open-source [ArduCopter](#) flight stack was flashed onto the FCU.

Amongst the safety features implemented using the ArduCopter firmware are geofencing, emergency motor stop, retractable landing gear and parachute deployment for safety,

as well as GNSS-RTK integration for enhanced precision and gripper integration readiness. Another feature of the ArduCopter firmware is [Mavlink](#) protocol integration, a simple to use protocol for mission planning and communication with aerial vehicles. The [Here+ RTK GNSS](#) receiver has been employed for reducing the measurements in altitude in 3 to 4 cm[11].

The attached Intel i7 enabled board allows the end-user to plan missions on-the-fly using geographic coordinates. The companion computer can be used for any visual-servoing; a custom casing for the NUC computer is designed for direct mounting on the the underside mounting rail of the octarotor, as shown in Figure 2 with a 450 g weight.



Fig. 2: Intel NUC mounting assembly

C. GNSS-RTK

The Global Navigation Satellite System (GNSS) - Real-time kinematic positioning (RTK) is a technique used to enhance the position's precision of satellite-based positioning systems such as GNSS, GLONASS, Galileo, and BeiDou. This can be archived with a combination of a fixed or moving base GNSS receiver (Base) which is used for reference, and one or more moving GNSS receivers (Rover). The Base and Rover are communicating with each other in real time, through a communication link. The Base re-broadcasts the phase of the carrier that it observes, and the Rover units compare their own phase measurements. This allows the Rover units to calculate their relative position to within millimeters. The GNSS receivers that have been chosen for the delivery drone are the u-blox NEO-M8P with typical nominal accuracy of 0.025m when the RTK is used "[GNSS receiver specifications](#)".

D. HD video transmitter

A [digital video link](#) supporting 1080p60 video resolution with 1 msec-latency, operating in the 5.1 up to 5.8 Ghz band with 40 MHz channels and relying on AES-128 encryption using RSA-1024, and LoS range up to 1 Km is used. This digital link, shown partially in Figure ??, provides accurate, encrypted and seamless video streaming from the attached RGB-D camera.

III. SAFETY FEATURE ENHANCEMENTS

The hardware enhancements include a protective bumper for the four coaxial pairs of propellers, a parachute, and two RGB-D cameras. The forward looking camera is used for obstacle avoidance in navigation while the downward facing assists in the feasible landing.

A. Protective propeller bumper design

The 3D-printed plastic Protective Propeller Bumpers (PPBs) have a 200 g weight (per coaxial pair of rotors), shown in Figure 3. The cover consists of lightweight (white colored) flexible plastic parts on the extremities so as to absorb energy from side impacts, as well as 10mm in diameter carbon tubes to enhance rigidity and stability during flight. Each bumper is mounted at the mounting plates of the coaxial motors. A lower crash structure can be visualized for absorbing energy in case of underside impacts with humans.

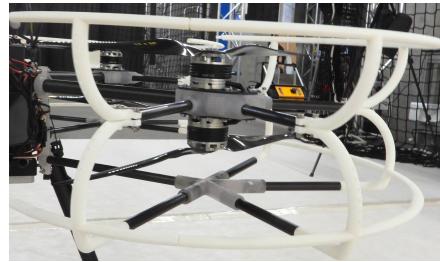


Fig. 3: Protective propeller bumper structure

These bumpers provide significant safety against direct forward collisions with obstacles, while in an emergency landing scenario while falling with the assistance of a parachute the PPB will crash and absorb energy thus avoiding direct contact of humans on the ground with the spinning blades.

Using a [FEA package](#), the PPB-structure was optimized, and can offer a stiffness of XX N/mm against a forward applied force, as shown in Figure 4. With a gap of YY mm between the bumper's circumference and the tip of the propellers this amounts to sustaining up to ZZ N forward force prior to crashing the propellers to the structure; this force is equivalent to a forward linear velocity of AA Km/h given the mass of the drone and the autopilot observes this upper limit.

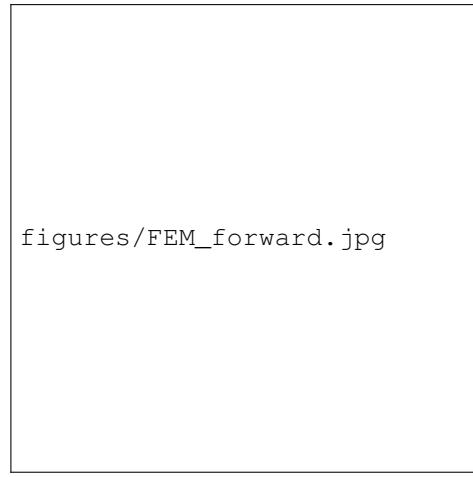


Fig. 4: FEM-stiffness for PPB (forward contact)

Similarly, in case of loss of power to the motors and the use of a parachute, resulting in a terminal descending velocity of 6.7 m/sec the carbon rods and the bumper will also absorb energy while the provided stiffness corresponds to BB N/mm,

shown in Figure ?? which is sufficient to avoid contact of the blades to the humans.

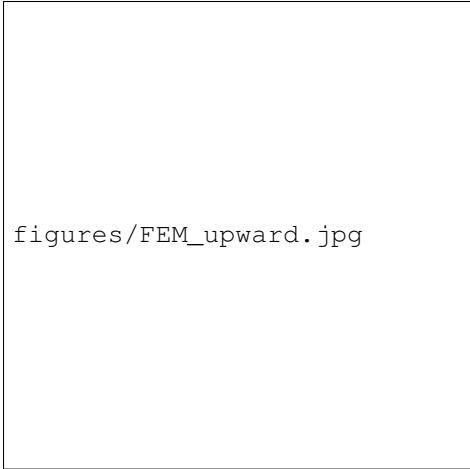


Fig. 5: FEM-stiffness for PPB (downward contact)

B. Parachute mechanism

A 48 inch [parachute](#) with an ejection mechanism of the shroud lines and its extension cord followed by the parachute is used, as shown in Figure 6. This parachute for a 7 Kgr drone offers a descent speed of 6.7 m/sec and an impact energy of 157 Joules (equivalent to a free fall from 2.26 m). An extra RC receiver is used to activate the trigger through an externally supplied PWM signal. The weight of the parachute is 250 g (excluding the additional 4S 1300 mAh LiPo battery), shown in Figure ??.



Fig. 6: Parachute and Video Transmitter

IV. ENVIRONMENT PERCEPTION

A. RGB-Depth camera and video transmission

The drone is equipped with an [Intel D435i RGB-D camera](#), which enables acquisition of aligned depth and RGB images; and provides this information based on an internal FPGA-circuit at 30 FPS. Having measured the aligned depth from all pixels, an obstacle classifier mechanism is enabled that provides a rectangular Bounding Box (BB) around the detected object, followed by an averaging of the pixels' depth within this box. To account for any inconsistencies, the averaging

mechanism is altered to a histogram-based inlier pixel classifier, where the histogram of distances is generated with a pre-determined resolution for the region within the BB. Consecutive triplets of bins are summed up, and the triplet with maximum number of summed counts is selected as the object location. Inlier pixels that are known to fall into the selected triplet are averaged to calculate distance to the object. Since the BB is also known, the relative location of the obstacle can be estimated using the camera intrinsics and the estimated distance to the object. A demonstration of the proposed pipeline is provided in Figure 7 with the identified BB of the balloon.

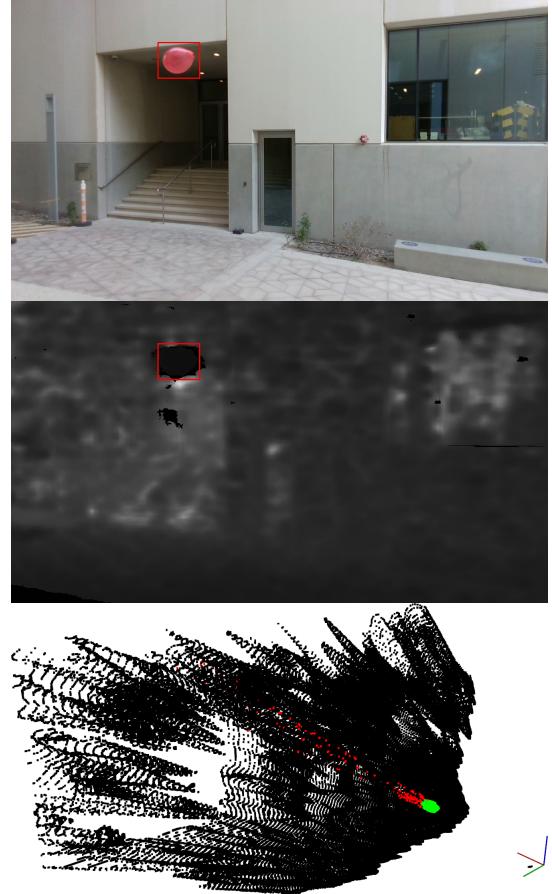


Fig. 7: Depth estimation pipeline: RGB-image[Top], Depth-image [Middle], Background rendering (black), outlier (red), and inlier (green) measurements [Bottom]

B. Deep learning for obstacle classification

A real-time deep learning based object detector is employed to visually detect obstacles similar to drones or balloons during the flight. This detector relies on the YOLOv4 tiny detector [12] optimized for the Intel NUC. Since it is difficult to obtain training datasets which include balloons and/or drones, a simulated dataset was employed that blends these objects with background images, as shown in Figure 8. In the 'collected' images, random augmentation (translation, scaling, rotation and distortion) is applied to enrich their divergence,

followed by the Poisson Image Blending method for generating synthetic datasets.



Fig. 8: Synthetic training image

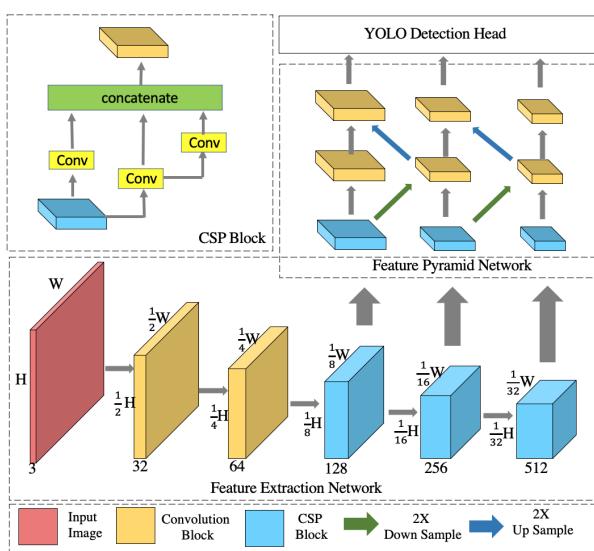


Fig. 9: YOLOv4 Tiny Object Detection Framework

The Yolo-detector shown in Figure 10 consists of: a) the extractor of the pyramid feature maps from images and b) the object classification and regression head. The pyramid feature maps with various scales are designed to detect objects of different sizes. These feature maps are further fused with each other to increase the detection performance. The final object classification and regression head outputs the location and BB size of the objects of interest.

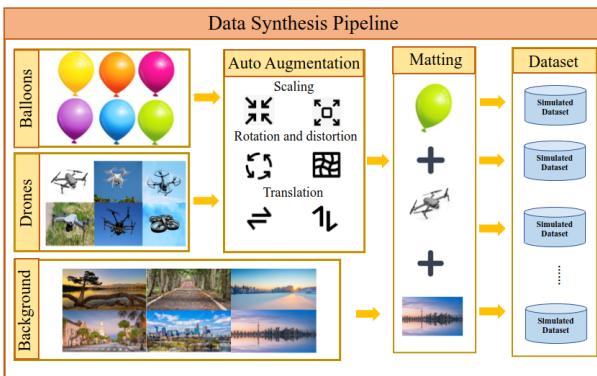


Fig. 10: Dataset synthesis pipeline

V. NAVIGATION

The navigation of the delivery drone, relies on the [Arducopter's flight controller algorithm](#) which is assigned with a series of waypoints to pass through. The flight controller algorithm receives the set position commands from the onboard companion computer through a serial wired or wireless UART port using the [Mavlink protocol format](#).

The object avoidance “Bendy ruler” code implemented within the Ardupilot is selected, where intermediate path waypoints are calculated either over or around the obstacle.

A. Drone landing

VI. PAYLOAD GRIPPER MECHANISM

A human operator loads the payload to the gripper mechanism, while the drop-off is automated. The final design used a dual linear actuator relying on the [Actuonix L-16 linear servo](#), with 63:1 gear ratio and a 100 mm stroke, being capable of grasping objects from 8 cm up to 30 cm. The power (6 V, up to 650 mA) is provided by the the drone’s power supply, while the maximum exerted force is 50 N. These PWM-controlled actuators can be controlled from a pre-programmed autopilot port for opening the gripper jaws during package drop-off.

The final payload handling mechanism is seen in the bottom image of Figure 11. In this design, by utilising the drone’s built-in payload mechanism of parallel carbon tubes, the use of an additional rail system was avoided. Each finger is equipped with linear bearings allowing it to seamlessly slide along the carbon tubes, while the actuators are housed in a separate part which is also clamped on the same carbon tubes.

Moreover, an internal lattice design was applied on the gripper fingers and their supporting extensions, thereby guaranteeing rigidity and strength while minimizing weight. This entire redesign led to the entire gripper weighing close to 350gr, with 150gr being the weight of the actuators. Despite this small weight, the gripper successfully managed to hold payloads of up to 2.5Kg in weight.

In order to minimize human interference, the gripper was also designed to be able to autonomously grasp packages. Since the selected actuators are not equipped with position feedback, the ACS723 current sensors were included in the controller, allowing for each finger to autonomously close until a high current reading is detected, indicating contact with the payload.

Although the bottom supporting extensions play an important role towards capably grasping the payload, the most important effect is the force normal to the payload’s surface, due to the gripper fingers. By adding a load cell on one of the fingers, this maximum gripping force could be measured. Figure 12 shows the filtered load cell readings from a total of 5 different scenarios.

As seen in the Figure, at the moment the controller reads a high-enough current and opts to stop its corresponding actuator, each gripper finger is applying close to 600gr of normal force.

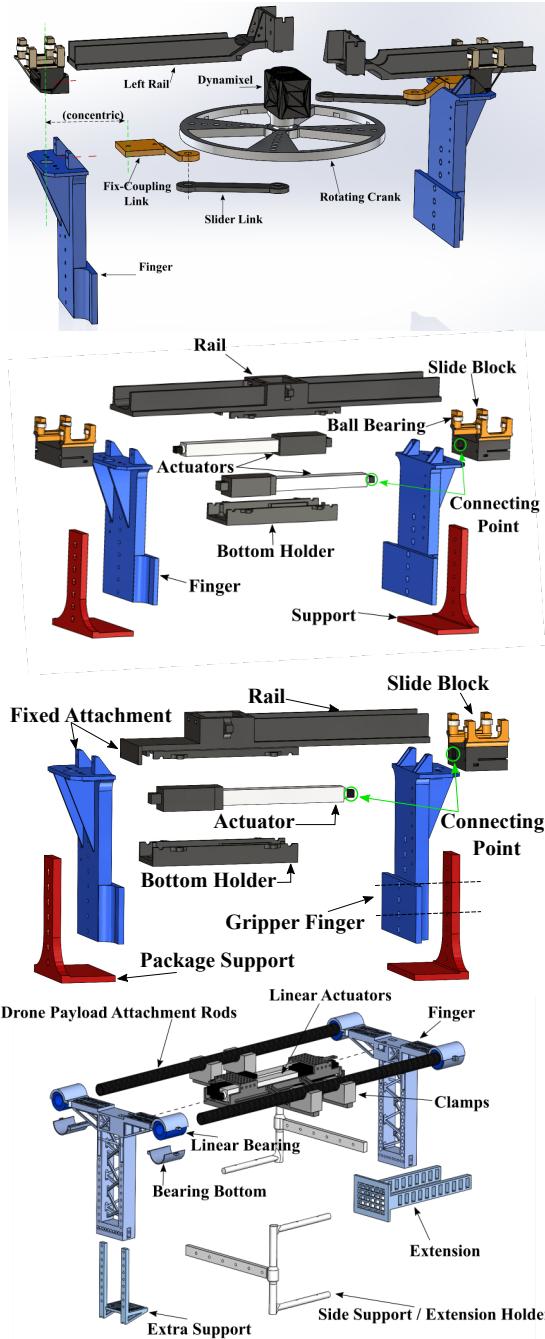


Fig. 11: Gripper mechanisms

VII. EXPERIMENTAL STUDIES

NYUAD's delivery drone participated as a finalist in the [Dubai World Challenge](#). The objective was to follow a 150 m path while carrying an unknown payload. The payload was $20 \times 10 \times 5$ cm with a 550 g weight. The objective was to move as close as possible to the path at a minimum time. There were balloons acting as obstacles and the drone had to avoid them by [passing either around or over them](#). At the same time, the drone's path was constrained to reside within a certain height and a narrow corridor and had to move under a building overpass. The trajectory consisted of 18-waypoints, shown in Figure 13, two balloons, one overpass and the drone

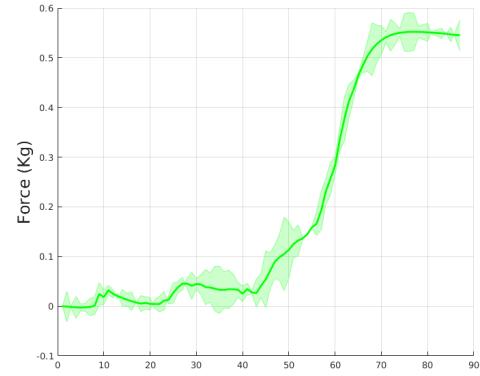


Fig. 12: Force applied by each gripper finger during grasping.

should land in a circle with radius 1.5 m; the overall time needed for our octarotor was 30 sec to complete the mission.

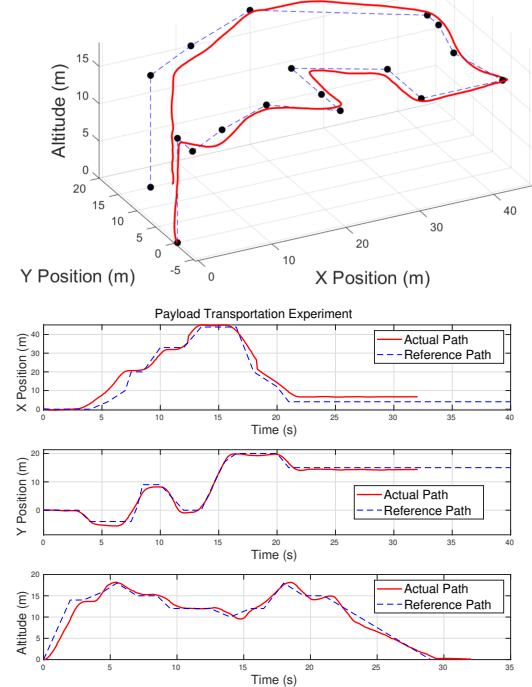


Fig. 13: Drone 3D-Trajectory

VIII. CONCLUSIONS

REFERENCES

- [1] D. K. Villa, A. S. Brandao, and M. Sarcinelli-Filho, "A survey on load transportation using multirotor UAVs," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, pp. 267–296, 2020.
- [2] M. Moshref-Javadi, S. Lee, and M. Winkenbach, "Design and evaluation of a multi-trip delivery model with truck and drones," *Transportation Research Part E: Logistics and Transportation Review*, vol. 136, p. 101887, 2020.
- [3] I. Hong, M. Kuby, and A. T. Murray, "A range-restricted recharging station coverage model for drone delivery service planning," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 198–212, 2018.

- [4] A. Tsoukalas, A. Tzes, S. Papatheodorou, and F. Khorrami, "UAV-deployment for city-wide area coverage and computation of optimal response trajectories," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 66–71.
- [5] E. E. Yurek and H. C. Ozmutlu, "A decomposition-based iterative optimization algorithm for traveling salesman problem with drone," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 249–262, 2018.
- [6] A. Gupta, T. Afrin, E. Scully, and N. Yodo, "Advances of UAVs toward Future Transportation: The State-of-the-Art, Challenges, and Opportunities," *Future Transportation*, vol. 1, no. 2, pp. 326–350, 2021.
- [7] M. Moshref-Javadi and M. Winkenbach, "Applications and research avenues for drone-based models in logistics: A classification and review," *Expert Systems with Applications*, vol. 177, p. 114854, 2021.
- [8] E. Frachtenberg, "Practical drone delivery," *Computer*, vol. 52, no. 12, pp. 53–57, 2019.
- [9] F. Schenkelberg, "How reliable does a delivery drone have to be?" in *2016 annual reliability and maintainability symposium (RAMS)*. IEEE, 2016, pp. 1–5.
- [10] A. Koubâa *et al.*, *Robot Operating System (ROS)*. Springer, 2017, vol. 1.
- [11] T. Baybura, İ. Tiryakioğlu, M. A. Uğur, H. İ. Solak, and Ş. Şafak, "Examining the accuracy of network RTK and long base RTK methods with repetitive measurements," *Journal of Sensors*, vol. 2019, 2019.
- [12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.