INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2020, VOL. 66, NO. 4, PP. 759–768
Manuscript received July 1, 2020; revised October, 2020. DOI: 10.24425/ijet.2020.134038

Research Studio for Testing Control Algorithms of Mobile Robots

Sebastian Dudzik, Piotr Szelag, and Janusz Baran

***Abstract*—In recent years, a significant development of technologies related to the control and communication of mobile robots, including Unmanned Aerial Vehicles, has been noticeable. Developing these technologies requires having the necessary hardware and software to enable prototyping and simulation of control algorithms in laboratory conditions. The article presents the Laboratory of Intelligent Mobile Robots equipped with the latest solutions. The laboratory equipment consists of four quadcopter drones (QDrone) and two wheeled robots (QBot), equipped with rich sensor sets, a ground control station with Matlab-Simulink software, OptiTRACK object tracking system, and the necessary infrastructure for communication and security. The paper presents the results of measurements from sensors of robots monitoring various quantities during work. The measurements concerned, among others, the quantities of robots registered by IMU sensors of the tested robots (i.e., accelerometers, magnetometers, gyroscopes and others).**

***Keywords*—mobile robots, UAV, MATLAB Simulink, QDrone**

I. Introduction

A. Mobile robots

Robotics [1] is a field of knowledge that has been developing very dynamically in recent decades. There are more and more new technical solutions that support or replace human work. This work is often done in difficult conditions or is physically heavy. One of the most dynamically developing branches of robotics are mobile robots, i.e. robots that can move in space. The space of movement is not limited only to land, the mobile robots can also fly and operate on or under water. These robots can be teleoperated by man or they can be autonomous units. The land-based robots, depending on the mobility system used, can be divided into: wheeled robots, crawlers, walkers and other [1]. Wheeled robots are used in many areas of human life starting from simple home applications (cleaning works, recreation), through education, industry (including transport [2], [3]) and ending in narrow, highly specialized works, e.g. defusing bombs or working in a contaminated environment. There are developed solutions whose goal is to improve communication between robots and their capabilities of autonomous work [3]. Equipping the robots with communication interfaces and a growing number of sensors and cameras, makes it possible to realize more advanced autonomous tasks as well as cooperation of robots. A robot must be equipped with appropriate tools to perform the tasks set for it. This is accomplished by a number of locators (GPS, laser, sonar) [5], [6], vision systems and other sensors. To move efficiently in the environment, the robot should have capability of creating a map of the surrounding space and locating itself in that space. These issues are very important research areas [4], [7]. The development of mobile robots is strictly connected with the invention of new materials, power systems, increasing the computing power of electronic systems and development of control algorithms. New solutions concerning localization and control (e.g. route planning) [8]–[10], [14], communication [13]–[16], autonomous operation [17]–[19], cooperation with other robots (e.g. formation control) [11], [12], [14], contribute to the emergence of better designs that are applied to ever new fields of human life.

Despite dynamic development of mobile robots technology, there are still many issues that need improvement be improved. In the authors' opinion, this applies in particular to the development of location and navigation algorithms for autonomously operating mobile robots. New control possibilities appear along with the emergence of more advanced design solutions that use better materials, sensors and cameras, communication methods or processors. New design solutions actually force development of more advanced systems of control or cooperation. An example of this are the algorithms that enable collaboration of mobile robots based on fast and reliable communication. Controlling swarms of mobile robots is currently an issue undergoing a rapid progress. The test stand presented in the article enables development and testing of more advanced algorithms for location and navigation of mobile robots. This applies to both single robot missions and robots that create swarms and cooperate with each other.

B. Drones a type of mobile robots

When discussing mobile robots one should note on dynamically developing flying robots, called unmanned aerial vehicles (UAVs) or simply drones. In the recent two decades, many new solutions have been developed to improve the control and functionality of drones, which has resulted in their widespread use and popularization, which was accompanied by a fall in prices. It is now very common to see operators performing recreational flights or making movies from above. It was necessary to establish laws regulating the conditions of flying.

The project financed under the program of the Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in the years 2019 - 2022 project number 020/RID/2018/19, the amount of financing 12,0000,00 PLN.

Authors are with Czestochowa University of Technology, Czestochowa, Poland (e-mail: sebdud@el.pcz.czest.pl, baranj@el.pcz.czest.pl, piotr.szelag@pcz.pl).

Due to the dynamic development in this field, these provisions are also subject to changes. UAVs are mainly associated with small vehicles flying in a strictly defined space (the flights are defined as within the visual line of sight or beyond the visual line of sight). However, big military UAVs can travel thousands of kilometers. They can be controlled by people or flight autonomously [20]. Depending on the purpose of the flight, there are very different sizes, equipment and constructions of drones. UAVs perform missions that are dangerous or difficult for humans. They are most often used for surveillance, searching, transport, security or rescue. There are currently many types and models of UAVs. Their classification is based on various parameters. The most popular parameter is drone weight. Taking the application as the criterion, the UAVs are divided into units for recreational, professional or industrial use. The main difference between them lies in the devices, systems and materials used for construction. Considering the number of rotors, the most popular solution are quadcopters with four arms and motors. Constructions with a larger number of arms, i.e. hexacopters or octocopter, allow for a more stable flight, and also ensure better control and safety in the case of failure of one of the rotors. Less common designs are AUVs with three (tricopter) or two (bicopter) rotors. The basic elements and systems used in drones are:

- frame - several basic constructions can be distinguished, the larger the number of arms, the more stable is the drone flight,
- propulsion system with a power source.
- propellers generating lift.
- flight control systems.
- position and flight sensors providing information about the status of the drone.
- software for control, communication or mission planning.
- operator ground station equipment for communication, control and displaying images from the drone camera(s).

C. Applications of UAVs

Depending on the equipment (sensors, cameras), construction and size, UAVs can be used for various missions. Types of missions can be divided into several categories, e.g.: civil or military, operation area (indoor, outdoor), type of environment (air, water). Common civil applications of drones include photography, film industry, natural disaster management, transport and agriculture [22]. One of important applications of drones are search missions, that may sometimes turn into rescue missions [21]. Drones allow for quick and efficient search of a specified area, using video or thermal cameras. They can be used to monitor the status of power networks, photovoltaic panels or for creation of orthophotomaps [23]. Patrol missions realized by drones allow monitoring of natural resources in areas difficult to access. These missions are especially helpful in locating forest fires or searching for missing persons. Finally, drones are employed in a very wide range of military applications.

Fig. 1. Drone with markers for identification and tracking

II. Methods

A. Description of the test stand in the Laboratory of Intelligent Mobile Robots

The test stand was built in the Laboratory of Intelligent Mobile Robots at the Faculty of Electrical Engineering of Czestochowa University of Technology. The main component of the laboratory equipment is the Autonomous Vehicles Research Studio provided by Quanser. The Quanser solution is a complete hardware and software setup that enables conducting research on autonomous wheeled robots (QBot) and flying robots (QDrone). The basic elements of the stand are:

- Ground control station including a PC computer with Intel Core i7 quad-core processor, three FullHD monitors, high performance Wi-Fi router, flight controller joystick and a ground camera
- Software: Matlab 2018a, QUARC Real-Time Control Software 2018, Motive 2.0, Visual Studio Community Compiler 2017.
- Mobile robots: four QDrone quadcopters, two QBot 2e 3-wheel robots.
- Optitrack Flex 13 localization system with eight fixed cameras for tracking positions of the mobile robots in 3D space. The cameras are mounted under the ceiling around the workspace: four are located in the corners, the other four in the middle of the sides.
- Workspace of size 5mx5mx2.5m in which the mobile robots can move, secured with a net, flexible mats (on the floor) and sponges (on the wall).

The Optitrack Flex 13 system uses dedicated optical motion recording software - Motive 2.0 (Optical motion capture software). The parameters of the tracking cameras are as follows:

- Resolution - 1280 x 1024,
- Native Frame Rate - 120Hz,
- Latency - 8.3ms,
- 3D Accuracy +/- 0.20 mm,
- Passive Markers - 9m,
- Stock Lens - 56o x 46o FOV,
- Shutter Default Speed - 0.25 ms

To make possible recognition and tracking, the drones have special reflective markers (small balls) installed on the top of

their frames. The layout of the markers has to be asymmetric and unique for a given drone so that the software could identify it and determine its position and orientation in space. Figure 1 shows a QDrone used in the tests, whose markers are marked with red circles. On the basis of the markers pattern the tracking software defines a 3D rigid body frame for each robot and further uses it for identification and tracking. Figure 2 shows the view of the drone from the Motive software perspective. The image on the top shows the set of markers before creating the rigid body, the image on the bottom shows the rigid body created from the markers.

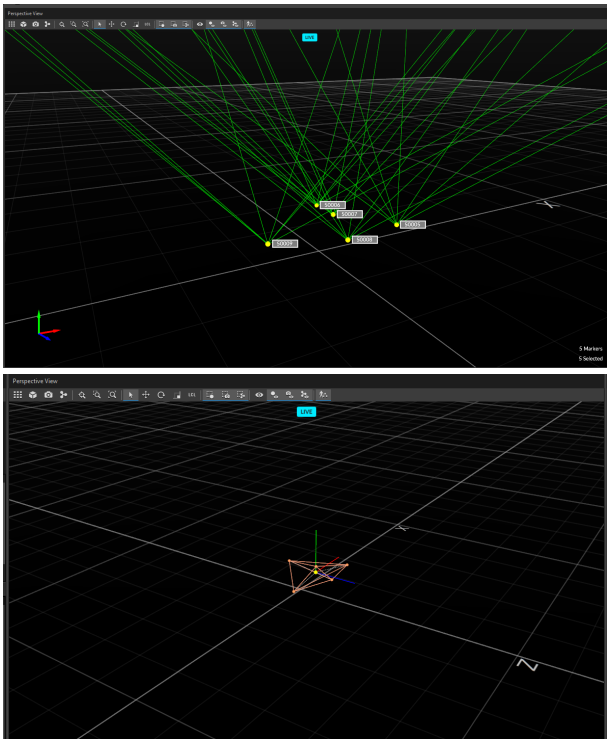

Fig. 2. Drone seen in the Motive software: set of markers (top) and the rigid body frame (bottom)

In mechanics the term rigid body is used for a solid body whose deformations are negligible or zero. Therefore, the distance between any two points of the rigid body remains constant, even when external forces are exerted on it. Due to this property, it is possible to determine the pose of a rigid body on the basis of positions of minimum three non-collinear points attached to this body. The markers attached to the robot define these points. The rigid body frame the marker points are used to determine the geometric center of mass or the centroid of the rigid body frame, which is the default position of the so called pivot point. The pivot with the markers constitute the rigid body frame which is used in the Motive software to specify position and orientation (pose) of an object in space. The location of the pivot point can be changed in the software because it is usually better to move it to the real center of the mass [24]. In our case, the five markers are attached asymmetrically to the upper part of the drone, mostly around its frame, because longer distances between the markers result in a larger rigid body frame and ensure the most accurate 3D orientation data of the object in the software. The recommended number of markers attached to an object is from 4 to 12. The markers are placed on the upper surface of the drone to ensure better visibility for the cameras installed under the ceiling but, in consequence, the default pivot point is above the real center of mass and has to be corrected. The information on the height of the drone mass center relative to its bottom base is given in the technical documentation so the vertical position of the pivot point can be determined easily. The location of the center of mass in the horizontal plane is a slightly bigger problem. This point is located almost at the center of the drone and marked by the manufacturer on the upper frame. In Figure 1 it is marked with a green circle. The horizontal position of the center of mass in the rigid body frame can be determined knowing the distance of this point from the markers and the coordinates of the markers. In practice, the procedure of the pivot point moving looks as follows: an additional marker ball is placed temporarily on the upper frame of the drone at the center of mass (in the horizontal plane). By reading the coordinates of the additional marker in the software, we obtain two of the three coordinates of the center of mass, the third (vertical) coordinate is taken from the technical documentation. With these values it is possible to move the pivot point from the default position to the actual center of mass of the drone. The process of moving the pivot point from the default location to the actual center of mass of the drone is shown in Figure 3. It shows the shift in the plane perpendicular to the image (front view) which involves modification of two coordinates, but the process of correcting the second horizontal coordinate in a perpendicular plane (side view) is the same. Figure 3a shows the rigid body before moving the pivot point. The additional marker with the pivot point already moved to the right place is shown in Figure 3b. Figure 3c shows the completed process the pivot point is moved to the right place and the additional marker is removed.

To perform flights within the workspace with good tracking accuracy, it is necessary to calibrate the Optitrack Flex camera system first. This process consists of two stages. At first the system has to determine how the cameras are located in the space in relation to each other. The manufacturer delivers a special tool T-shaped calibration wand (CW - 500) with three suitably mounted markers. The markers size and distances to each other on the wand are stored in the Motive software. The wand has to be waved by an operator over the whole workspace and the positions of the wand markers seen by at least four cameras are recorded. Based on these observations, Motive carries out geometrical calculations and determines locations of the cameras relative to each other. To obtain accurate results the system requires at least 5000 recorded positions of the wand markers. The second step of the calibration is to determine positions of the cameras relative to the floor level. Another special tool with suitably mounted marker is used at this step - calibration ground square (CS - 200). As before, Motive knows the tool markers size and distances to each other. The calibration square is put on the floor in the middle of the workspace and information on the location of markers on the tool, seen at a different distance and

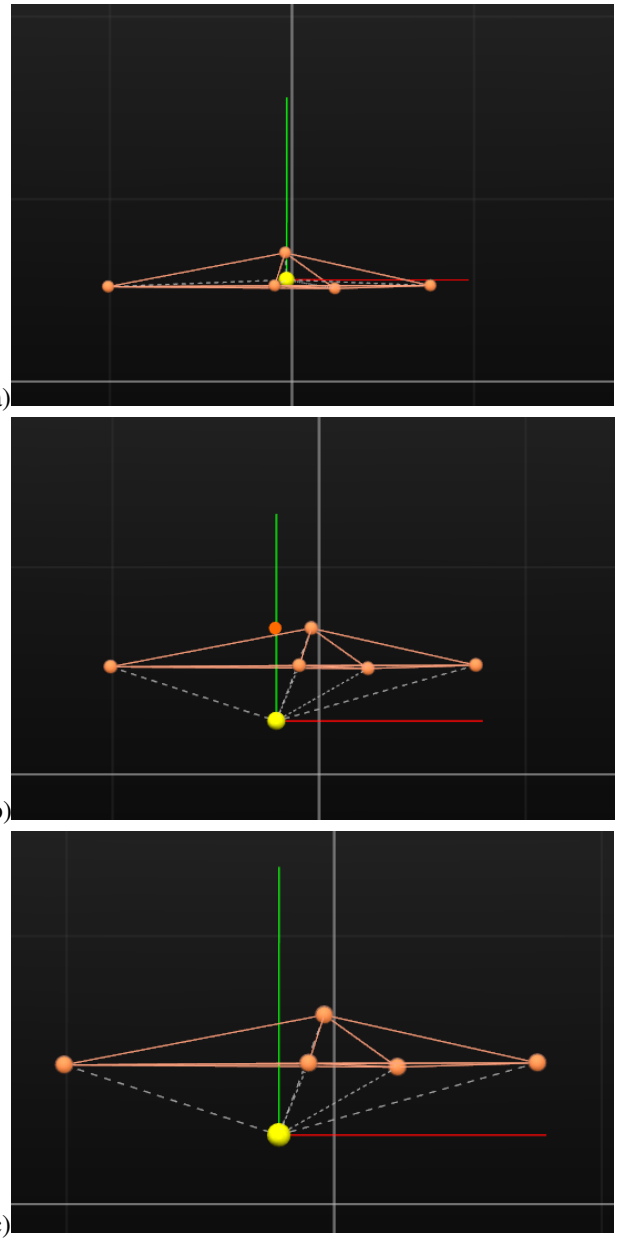

Fig. 3. Process of moving the pivot point from the default location at the geometric center of mass to the actual center of mass of the drone, front view

angle by different cameras is used by the software to determine the positions of the cameras relative to the ground.

As the result of the described definition of the rigid body (or more rigid bodies) and the calibration process, the software creates corresponding configuration files. One of them stores information about the system calibration, the other contain data about the rigid bodies of the mobile robots. It should be noted that due to changing environmental conditions (mainly related to the intensity and type of lighting) it is recommended to calibrate the system each time before starting tests on a given day. It requires some time, because only the warmup of the cameras takes about 20 minutes. The configuration files are necessary to perform missions supervised by algorithms run in the Matlab/Simulink environment with the QUARC real time software. These files are input data to another component the laboratory system, i.e. the Optitrack Trackables, which returns positions of objects being tracked by the OptiTrack camera system.

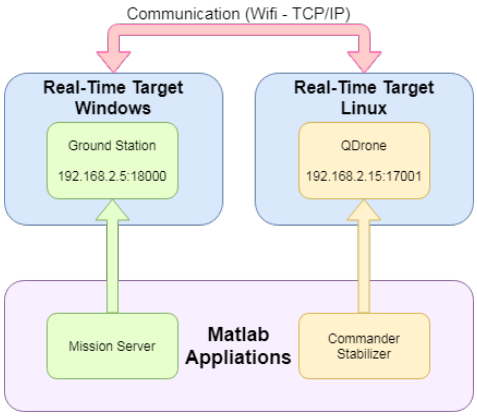

Fig. 4. Diagram of the system software components and data exchange between the two target platforms: the ground station with Windows and the QDrone with Linux

Safe operation of a QDrone requires suitable preparation of software that supervises communication between the drone (or drones), the ground station and the flight controller joystick. The manufacturer provided sample modules and algorithms that allow for the above functionality. These basic software modules can be expanded by the user or he can create his algorithms from scratch, using only blocks available in the Matlab/Simulink libraries. The Simulink block that supervises communication with the flight controller is named Joystick. It receives data from the flight controller and passes it to other components of the Simulink flight control program. The QDrone flight control software consists of two main components named:

- Commander Stabilizer - QDrone
- Mission Server QDrone Position Control

These programs are compiled and built on PC in Simulink with QUARC and sent to one of the two target platforms shown in Figure 4. The Commander Stabilizer is run on the QDrone as a real-time Linux task, while the Mission Server is executed on the PC ground station as a real-time Windows task. Information about where and how the target program works is entered through the Simulation− >Model Configuration Parameters− >Code Generation section. In this section the user defines the system target file and the communication interface parameters, e.g. IP address of the ground station and a QDrone (each QDrone and Qbot has its own IP) and numbers of communication ports for data exchange between targets. The data exchange between the targets is realized by the Communication subsystem (see Figure 5).

The Mission Server system (Figure 5) includes subsystems responsible for communication with the flight controller joystick, localization of a drone (or drones), trajectory planning and saving the mission data. They are included in the block Mission Server QDrone Position Control. Tracking position of the drones during the mission is carried out by the Data Server

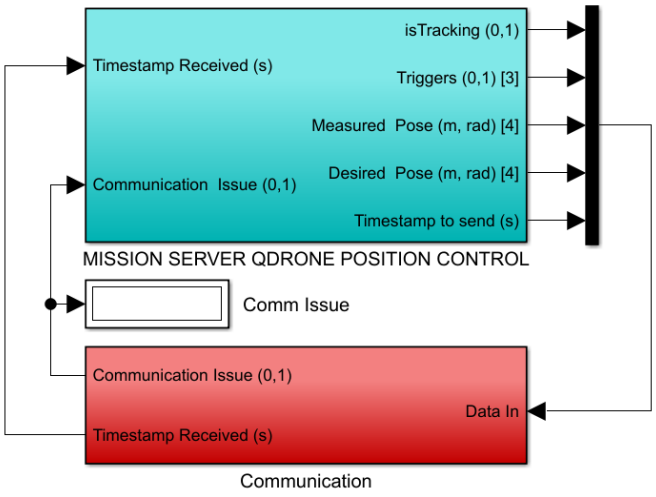

Fig. 5. Mission Server Application. Source: [25]

subsystem, whose task is to retrieve the position of the object being tracked by the OptiTrack cameras (OptiTrack Trackables block) and to convert the position data from the Motive Frame to the Research Studio Frame (these coordinate frames are different). Object tracking requires entering locations of the calibration and the object rigid body files along with its ID in the Motive software into the OptiTrack Trackables block.

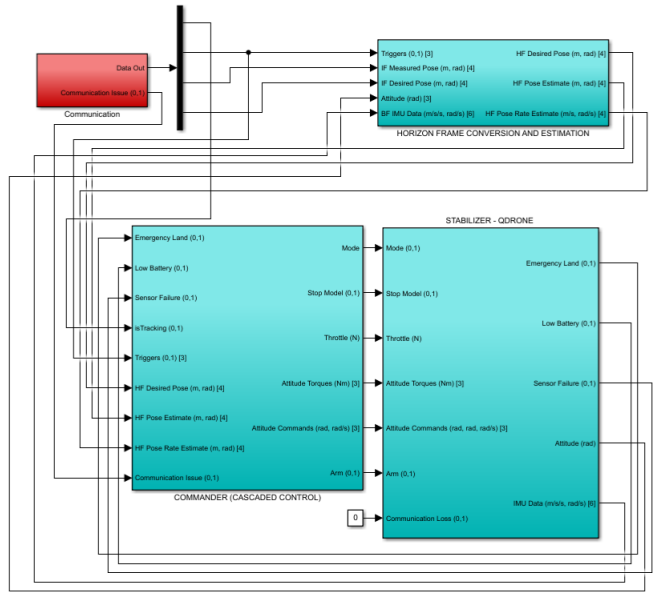

Fig. 6. Commander Stabilizer Application. Source: [25]

The Commander Stabilizer system includes three main subsystems (Figure 6):

- Horizon Frame Conversion and Estimation its task is to evaluate the Yaw Angle from the drone gyroscope signals, estimate the roll and pitch angles, transform the Body Frame matrix into the Horizon Frame matrix (acceleration components), the Inertial Frame matrix into the Horizon Frame matrix (positions and angles), as well as to estimate the Horrizon Frame Pose Rate (linear and angular velocities).
- Commander its task is to monitor the QDrone parameters using the Finite State Machine (FSM), adjust the desired position to the controller based on the FSM state, determine the forces acting on the object and adjust commands (Throttle Signals) to these forces.
- Stabilizer its task is the dynamic control of the QDrone; it allows to change the stabilization mode (Torque / Angle based), implement a cascade PIV controller, monitor the status of the module (communication with the ground station), implement I / O with QDrone motors, LEDs, sensors, evaluate the position.

For security reasons, one should at first build the Mission Server Simulink module and run it as a Windows target. After starting, the Mission sever connects with the tracking software and after a few seconds it should return a message about correct operation of the tracking system (Is tracking property will change to true). Only after that one can build, send to the QDrone target and run the Commander Stabilizer module. Correct operation of this module is signaled by audio and LED signals from the QDrone. At the same time, a set of graphs showing selected flight parameters is shown on the screen of the earth station. These flight data are also saved on the ground station's disk with a frequency of 1kHz.

B. QDrone

The QDrone flying robot used for tests is a 40x40x15cm quadcopter with propellers in X-configuration (see Figure 1). Its weight with the battery is about 1100g. The 3S 11.1V LiPo battery allow for about 11 minute long flights. Basic components of the QDrone are:

- Intel Aero Compute computer (Intel Atom x7-Z8750 quad-core 64-bit 2.56 GHz processor, 4-GB LPDDR3-1600 RAM, IMU (inertial measurement unit): BMI 160 with 6- DOF 16-bit triaxial accelerometer and gyroscope, BMM 150 magnetometer with 3-axis geomagnetic sensor, MS5611 barometer with 24-bit pressure and temperature sensor)
- Four Cobra 2100Kv (size 2206) motors: 2100 RPM / V, Max continuous current - 25 Amps.
- Dual-blade polycarbonate 6045 propellers: Diameter - 6.0 inches, Pitch - 4.5 inches).
- Single pack Lithium-Polymer (LiPo) 3S 3300mAh battery (Weight - 230g, dimensions (LxWxH) - 135 x 44 x 17mm).
- RGB-D Camera - Intel RealSense R200 (Depth Sense: 0.5 - 3.5 m range, IR Image: 480 x 320 at 60 FPS, RGB Vision: 640x480 at 60 FPS / 1080p at 30 FPS).
- Optical Flow Camera - Omnivision OV7251 (Output: grayscale, Vision: 640x480 at 120 FPS)

The main elements of the QDrone are indicated in Figure 7.

C. Research metodology

The results presented in this work were obtained from test flights along three simple trajectories generated using the

Fig. 7. Main elements of QDrone. Source: [26].

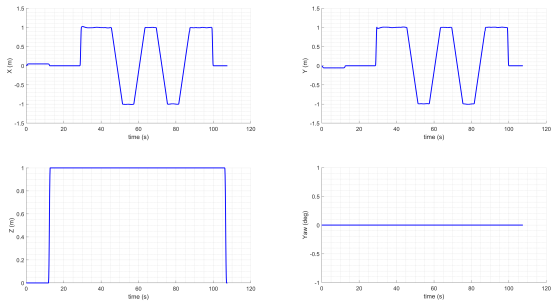

Fig. 8. Plots of desired coordinates and the yaw angle of the drone on trajectory T_1

waypoints mode. In this mode the operator creates a trajectory by defining successive desired poses of the drone in space in given time steps. The drone stays for a predetermined period of time in each of these poses. The trajectory programmed in the waypoint mode is described as the matrix of poses P and the vector of times τ:

$$\mathbf{P} = \begin{pmatrix} x_1 & y_1 & z_1 & yaw_1 \\ x_2 & y_2 & z_2 & yaw_2 \\ \vdots & \vdots & \vdots & \vdots \\ x_k & y_k & z_k & yaw_k \end{pmatrix}, \tau = \begin{pmatrix} \tau_1 & \tau_2 & \dots & \tau_k \end{pmatrix}, \tag{1}$$

where:
x_i, y_i, z_i, [m] – desired position coordinates for the i-th pose,
yaw_i, [rad] – desired attitude for the i-th pose described using the yaw angle (rotation of the drone around an axis perpendicular to the ground plane, the pitch and roll angles are set to be zero at the steady state),
τ_i, [s] – time, for which the drone remains in the i-th pose,
k – number of poses creating the trajectory.

The research consisted of the following stages:

- Programming the drone mission using specialized blocks of Simulink, including the Mission Server block. The elements of the pose matrix and the time vector (1) are determined at this stage,
- Take-off from a point close to the origin of the coordinate system and hovering at a height of 1 m.
- Start of the mission. The drone flies to the first waypoint (the desired coordinates and orientation are in the first row of matrix P) and stays at that point for the specified time (the first element of vector τ).
- Flights between successive waypoints in the workspace specified in matrix P. The time periods specified as successive elements of vector τ are allocated for each flight. This stage is continued until completion of the trajectory, i.e. until the drone achieves the last waypoint described by the k-th row of matrix P and until the k-th time period passes. At this stage, the ground server (Mission Server) and the OptiTrack system are carrying out real-time recording of the flight parameters transmitted from the drone, including the actual position of the drone, its speed and acceleration (linear and angular) measured by the IMU, as well as internal values monitored by the drone control system (e.g. the battery voltage).
- Return to the starting point and landing.
- Analysis of the recorded flight parameter results.

Formulas (2) and (3) show matrix of poses P_1 and time vector τ_1 that define trajectory T_1. Fig. 8 shows plots of the desired coordinates and the yaw angle of the drone on this trajectory versus time. The constant levels and their lengths correspond to the values specified in matrix (2) and vector (3) for the trajectory.

$$\mathbf{P}_1 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ -1 & -1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ -1 & -1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \tag{2}$$

$$\tau_1 = \begin{pmatrix} 2 & 12 & 12 & 12 & 12 \end{pmatrix}. \tag{3}$$

The waypoints of a bit more complicated trajectory T_2 are specified by matrix of poses P_2 (4) and time vector τ_2 (5). The time plots of the desired spatial coordinates of the drone moving along this trajectory are shown in Figure 9.

$$\mathbf{P}_2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1.3 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 1.3 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1.3 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 1.3 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \tag{4}$$

$$\tau_2 = \begin{pmatrix} 2 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{pmatrix} \tag{5}$$

The third defined trajectory T_3 is represented by matrix of poses P_3 (6) and time vector τ_3 (7) and Figure 10 shows the time plots of the desired spatial coordinates of the drone for this trajectory.

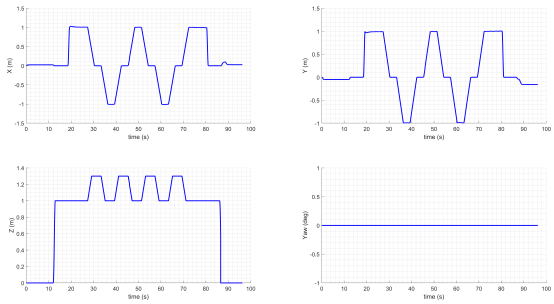

Fig. 9. Plots of desired coordinates and the yaw angle of the drone on trajectory T_2

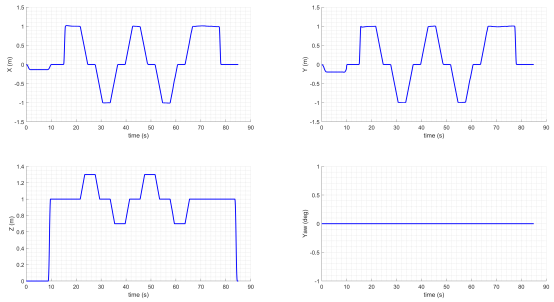

Fig. 10. Plots of desired coordinates and the yaw angle of the drone on trajectory T_3

$$\mathbf{P}_3 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1.3 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 0.7 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1.3 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 0.7 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \tag{6}$$

$$\tau_3 = \begin{pmatrix} 2 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{pmatrix}. \tag{7}$$

The following flight parameters, measured by the IMU module, were recorded in real time when the drone was following each trajectory:

- desired and measured position coordinates x, y, z,
- components of linear and angular velocity,
- components of linear and angular acceleration.

The rest of the work will present the results of research carried out in accordance with the methodology described above.

III. Results and Discussions

The records of flight parameters obtained for the three trajectories described above are presented below. The results regarding trajectory T_1 are presented in Figures 11-15. Time plots of the desired and the recorded - by the OptiTrack system

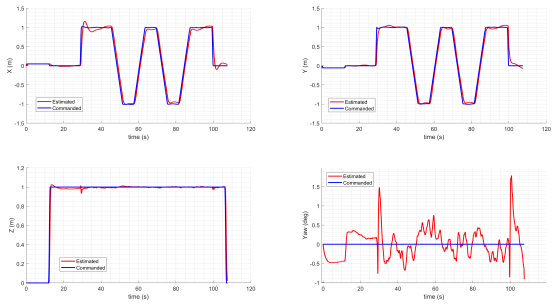

Fig. 11. Desired and recorded coordinates of the drone following trajectory T_1

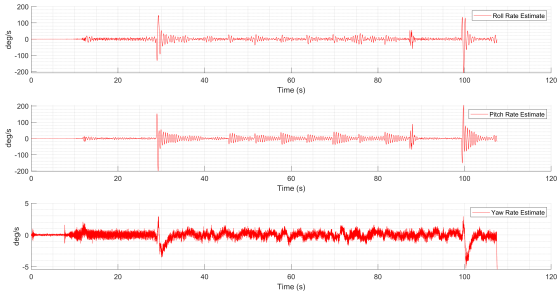

Fig. 12. Angular velocities in roll, pitch and yaw axes on trajectory T_1 measured by the drone gyroscope

- spatial coordinates of the drone following trajectory T_1 are shown in Figure 11. Figure 12 shows plots the drone angular velocities in the roll, pitch and yaw axes measured by the gyroscope built in the drone IMU unit and Figure 13 shows plots of the corresponding angular accelerations estimated from the angular velocities. Plots of the linear accelerations along x, y and z axes measured by the IMU accelerometer are presented in Figure 14 and the corresponding linear velocities estimated from the linear accelerations are shown in Figure 15.

The flight parameters obtained from the IMU measurements when the drone was following trajectory $T2$ are presented in Figures 16-20. The recorded and the desired spatial coordi-

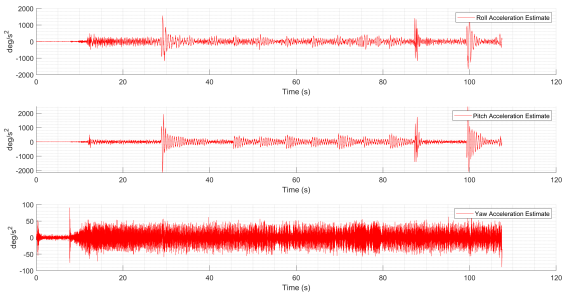

Fig. 13. Angular accelerations in roll, pitch and yaw axes on trajectory T_1 estimated estimated from the corresponding angular velocities

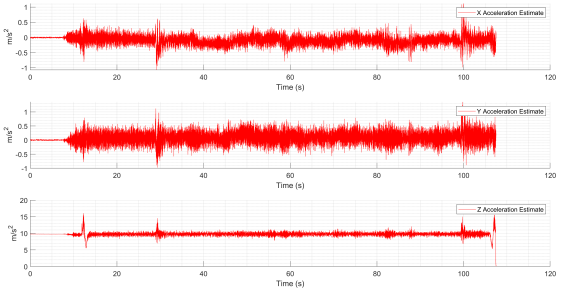

Fig. 14. Linear accelerations along x, y and z axes on trajectory T_1 measured by the drone accelerometer

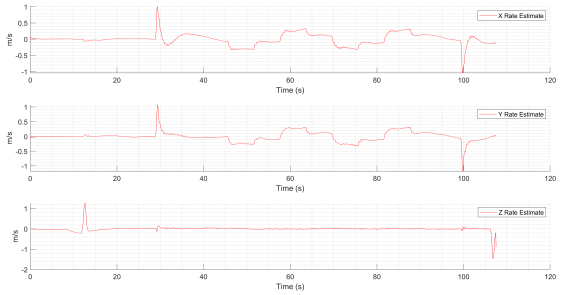

Fig. 15. Linear velocities along x, y and z axes on trajectory T_1 estimated from the corresponding linear accelerations

nates of the drone are plotted in Figure 16. Figure 17 shows components of the measured angular velocities and Figure 18 shows the corresponding estimated angular accelerations. Components of the measured linear accelerations are presented in Figure 19 and the corresponding estimated linear velocities are shown in Figure 20.

Finally, Figures 21-25 present, as before, the flight parameters obtained from the IMU measurements when the drone was following trajectory T_3. The recorded and the desired spatial coordinates of the drone are shown in Figure 21. Figure 22 presents components of the measured angular velocities and Figure 23 presents the corresponding estimated angular accelerations. Figures 25 and 25 show components of the

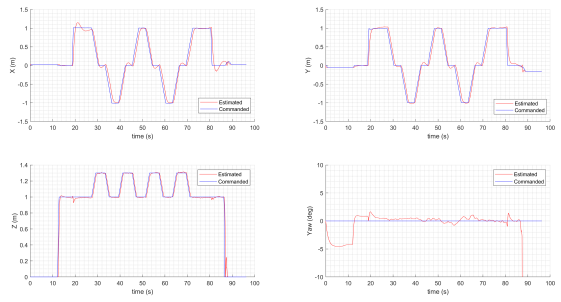

Fig. 16. Desired and recorded coordinates of the drone following trajectory T_2

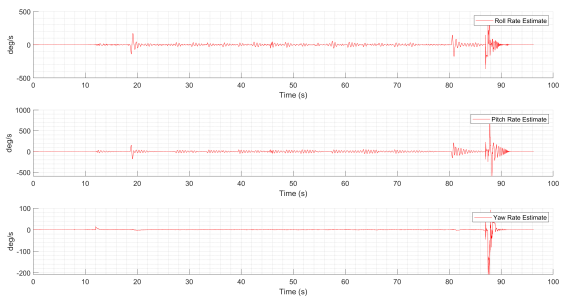

Fig. 17. Angular velocities in roll, pitch and yaw axes on trajectory T_2 measured by the drone gyroscope

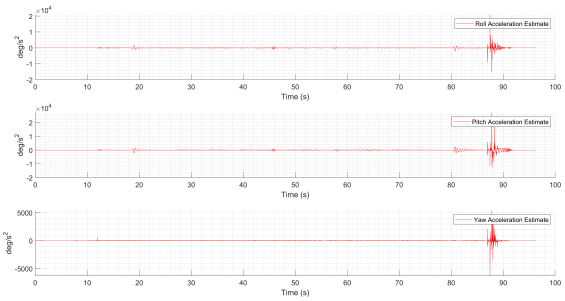

Fig. 18. Angular accelerations in roll, pitch and yaw axes on trajectory T_2 estimated estimated from the corresponding angular velocities

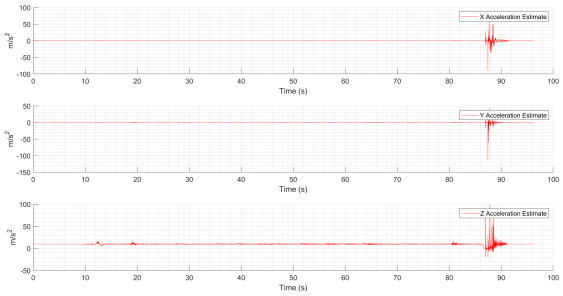

Fig. 19. Linear accelerations along x, y and z axes on trajectory T_2 measured by the drone accelerometer

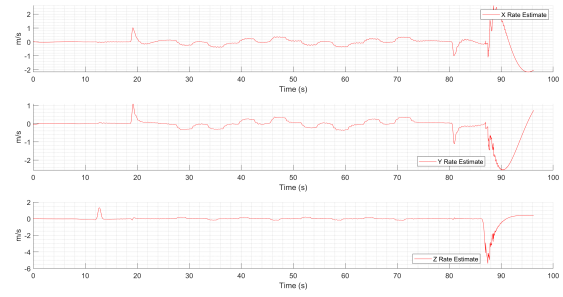

Fig. 20. Linear velocities along x, y and z axes on trajectory T_2 estimated from the corresponding linear accelerations

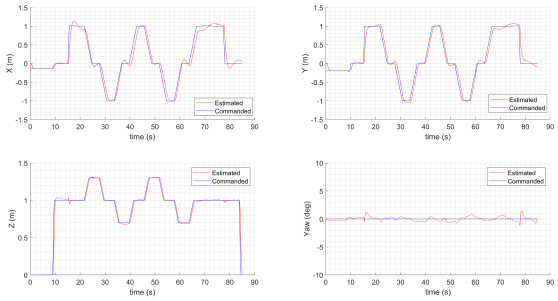

Fig. 21. Desired and recorded coordinates of the drone following trajectory T_3

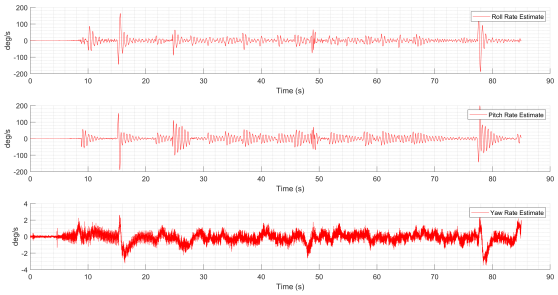

Fig. 22. Angular velocities in roll, pitch and yaw axes on trajectory T_1 measured by the drone gyroscope

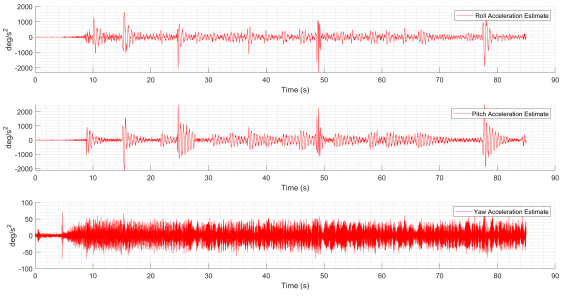

Fig. 23. Angular accelerations in roll, pitch and yaw axes on trajectory T_3 estimated estimated from the corresponding angular velocities

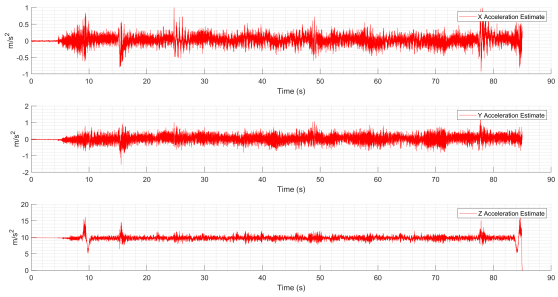

Fig. 24. Linear accelerations along x, y and z axes on trajectory T_1 measured by the drone accelerometer

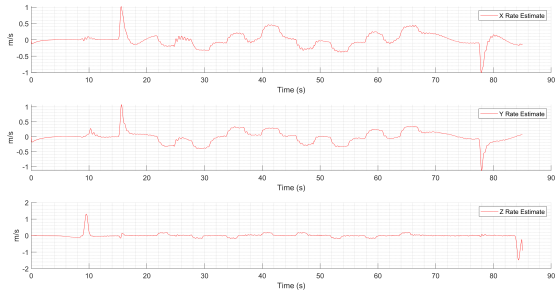

Fig. 25. Linear velocities along x, y and z axes on trajectory T_3 estimated from the corresponding linear accelerations

measured linear accelerations and the corresponding estimated linear velocities respectively.

The plots of the desired and recorded coordinates and yaw angle of the drone in Figures 11, 16 and 21 show that the presented laboratory stand makes it possible to investigate tracking accuracy for a programmed flight trajectory and improve control algorithms of drones (the QDrone in this case). For example, on the time plot in Fig. 11 we can observe fluctuations of the yaw angle resulting from unbalanced reaction moments around the z axis, which, in turn, results from the specifics of the control algorithm used to follow a given trajectory. Based on the plots in Figure 21 it is possible to evaluate the difference between the desired and the actual trajectory for individual spatial coordinates (for example, one can see about 15% overshoot of the position for the x coordinate).

In addition, as shown on the plots in Figures 12-22 and 15–14, the stand allows for real-time measurements and estimation of quantities such as components of the drone linear and angular velocities as well as components of the linear and angular accelerations. For example, in Fig. 15 one can observe changes in the profile of individual components of the linear velocity of the drone following trajectory T_1. Looking at the velocity component in the z direction, we can be seen that for most of the time it is approximately zero (the drone does not move vertically). The exceptions are time intervals representing the take-off (speed increases to approx. 12 m / s) and landing (speed increases to approximately -14 m / s - a negative value means the drone is moving down).

Finally, it should be noted that the results presented in Fig. 11-24 relate to simple trajectories generated using the waypoint mode, but the presented stand makes it also possible to test complex trajectories and advanced route planning algorithms.

IV. Conclusion

Algorithms related to drone control, navigation or location are constantly evolving. Due to growing popularity of drones, these research and application areas are becoming increasingly important, if only because of the safety issues associated with the use of drones. Therefore, it is important to ensure

efficient communication and reliable control algorithms on the one hand, and safety of humans and property, that could be destroyed as a result of malfunction control or positioning errors, on the other hand. Another important issue is design and implementation of missions beyond eye contact of the operator and the drone. In this case the reliability of control, navigation and location algorithms becomes even more important. In addition, it is necessary to take into account problems associated with trajectory generation, for example, by setting via points for used automatic calculation the drone flight.

This work presents the laboratory stand whose research capabilities fit very well into current needs. The software available on the stand makes it also possible to create mathematical models od drones of various properties and build and test algorithms for these models. The stand software allows the researcher to modify and test control structure and parameters (Matlab/Simulink), track the drone(s) in the laboratory workspace (OptiTrack cameras and software) and record flight data from the drone IMU (or other sensors that can be installed). It is carried out indoor in safe conditions. Testing of new solutions can be divided into two-step verification process. In the first step, the new solution can be tested on the software simulator. If it passes the tests, then the new algorithm (or the old algorithm with new parameters) can be implemented and verified using the drones, but still in laboratory conditions. In addition, the test stand enables simulation of various flight conditions, e.g. wind gusts.

The work presents also example test results obtained for three drone flight trajectories. The time plots of the desired (programmed) and measured spatial coordinates of the three trajectories are shown in Figures 11, 16 and 21. The differences between the programmed and actual drone flight path is visible. Apart from the tracking data, the recorded and estimated data related to flight dynamics, i.e. the components of the angular and linear velocities and accelerations, are also presented. The time plots of these quantities are presented individually for each trajectory and each coordinate axis.

The presented research stand allows to define strictly quantitative criteria of the control quality and compare them for various algorithms. Thus, it is possible to assess algorithms of control, navigation and tracking on the basis of data obtained from the drone following any trajectory.

REFERENCES

[1] M. Trojnacki, P. Szynkarczyk, A. Andrzejuk, "Tendencje rozwoju mobilnych robotów l¹dowych", *Pomiary Automatyka Robotyka* 6, 11–14 (2008).

[2] T. Machado, T. Malheiro, S. Monteiro, W. Erlhagen, E. Bicho, "Attractor dynamics approach to joint transportation by autonomous robots: theory, implementation and validation on the factory floor", *Autonomous Robots* 43, 589–610(2019), doi:10.1007/s10514-018-9729-2.

[3] H. Yamaguchi, A. Nishijima, A. Kawakami, "Control of two manipulation points of a cooperative transportation system with two car-like vehicles following parametric curve paths", *Robotics and Autonomous Systems* 63, 165178(2015), doi:10.1016/j.robot.2014.07.007.

[4] L. Paya, A. Gil, O. Reinoso, "A State-of-the-Art Review on Mapping and Localization of Mobile Robots Using Omnidirectional Vision Sensors", *Journal of Sensors*, vol 2017, 1-20(2017), doi:10.1155/2017/3497650.

[5] O. Wijk, H. Christensen, "Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data", *Robotics and Autonomous Systems*, 31(1-2), 3142(2000), doi:10.1016/S0921-8890(99)00085-8.

[6] A. Y. Hata, D. F. Wolf, "Outdoor mapping using mobile robots and laser range finders", in Proc. of the Electronics Robotics and Automotive Mechanics Conference, Cuernavaca, 2009, pp. 209214, doi: 10.1109/CERMA.2009.12.

[7] B. Madhevan, M. Sreekumar, "Identification of probabilistic approaches and map-based navigation in motion planning for mobile robots", *S?dhan?*, 43, 1-18(2018), doi: 10.1007/s12046-017-0776-8.

[8] G.C.R. De Oliveira, K.B. de Carvalho, A.S. Brand?o, "A Hybrid Path-Planning Strategy for Mobile Robots with Limited Sensor Capabilities", *Sensors* 19(5), 1049, 2019, doi:10.3390/s19051049.

[9] J. Bae, W. Chung, "Heuristics for Two Depot Heterogeneous Unmanned Vehicle Path Planning to Minimize Maximum Travel Cost", *Sensors* 19(11), 2461, 2019, doi:10.3390/s19112461.

[10] A. S. Brand?o, M. Sarcinelli-Filho, R. Carelli, "An Analytical Approach to Avoid Obstacles in Mobile Robot Navigation", *International Journal of Advanced Robotic Systems*, 10, 278, 2013, doi: 10.5772/56613.

[11] Z. Liu, W. Chen, J. Lu, H. Wang, J. Wang, "Formation Control of Mobile Robots Using Distributed Controller With Sampled-Data and Communication Delays," in IEEE Transactions on Control Systems Technology, vol. 24, no. 6, pp. 2125-2132, Nov. 2016, doi: 10.1109/TCST.2016.2518618.

[12] T. T. Ribeiro, R. O. Fernandez, A. G. S. Conceiç?o, "NMPC-based Visual Leader-Follower Formation Control for Wheeled Mobile Robots", IEEE 16th International Conference on Industrial Informatics (INDIN), Porto, 2018, pp. 406-411, doi: 10.1109/INDIN.2018.8472107.

[13] W. Li, L. Ding, H. Gao, M. Tavakoli, "Haptic Tele-Driving of Wheeled Mobile Robots Under Nonideal Wheel Rolling, Kinematic Control and Communication Time Delay", in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 1, pp. 336-347, Jan. 2020, doi: 10.1109/TSMC.2017.2738670.

[14] S. Rahmanpour, R. M. Esfanjani, "Energy-aware planning of motion and communication strategies for networked mobile robots",*Information Sciences*, 497, 149-164 (2019), doi: 10.1016/j.ins.2019.05.034.

[15] J. Fink, A. Ribeiro , V. Kumar, "Motion planning for robust wireless networking", in: 2012 IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, Minnesota, USA, 2012, pp. 24192426, doi: 10.1109/ICRA.2012.6224725.

[16] Y. Kantaros, M.M. Zavlanos, "Distributed communication-aware coverage control by mobile sensor networks", *Automatica* 63, 209220 (2016), doi: 10.1016/j.automatica.2015.10.035.

[17] F. Ribeiro, G. Lopes, T. Maia, H. Ribeiro, P. Osorio, R. Roriz, N. Ferreira, "Motion Control of Mobile Autonomous Robots Using Nonlinear Dynamical Systems Approach", in CONTROLO 2016 Proceedings of the 12th Portuguese Conference on Automatic Control, Springer International Publishing, 409–421, 2017, doi: 10.1007/978-3-319-43671-5_35.

[18] M. Pandelea, L. Vladareanu, M. Iliescu, R. I. Munteanu, M. Radulescu, "Intelligent Advanced Control Strategies for Mobile Autonomous Robots Stability Through Versatile, Intelligent, Portable VIPRO Platform," The 42th International Conference on ICMSAV, pp.184-189, 2018.

[19] M. Ghaffari Jadidi, J. Valls Miro, G. Dissanayake, "Gaussian processes autonomous mapping and exploration for range-sensing mobile robots" *Autonomous Robots* 42, 273290 (2018), doi: 10.1007/s10514-017-9668-3.

[20] M. Hassanalian, A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review", *Progress in Aerospace Sciences* 91, 99-131(2017), doi:10.1016/j.paerosci.2017.04.003.

[21] S. Waharte, N. Trigoni, "Supporting Search and Rescue Operations with UAVs," 2010 International Conference on Emerging Security Technologies, Canterbury, 2010, pp. 142-147, doi: 10.1109/EST.2010.31.

[22] G. Singhal, B. Gaurav, L. Mathew, "Unmanned Aerial Vehicle Classification, Applications and Challenges: A Review", Europe PMC, 2018, doi: 10.20944/preprints201811.0601.v1

[23] H. González-Jorge, J. Martínez-Sánchez, M. Bueno, "Unmanned aerial systems for civil applications: A review", *Drones*, vol. 1, p. 2, 2017, doi:10.3390/drones1010002.

[24] $https : //v22.wiki.optitrack.com/index.php?title = Rigid_Body_Tracking$

[25] $https : //www.quanser.com/products/autonomous - vehicles - research - studio/$

[26] $https : //www.quanser.com/products/qdrone/$