
COMPUTATIONAL AND DATA ACQUISITION SYSTEMS

Algorithms and Architecture of the Multirotor Aircraft Trajectory Motion Control System

K. Yu. Kotov^{1*}, A. S. Mal'tsev¹, A. A. Nesterov¹, and A. P. Yan^{1**}

¹*Institute of Automation and Electrometry, Siberian Branch, Russian Academy of Sciences,
Novosibirsk, 630090 Russia*

Received December 30, 2019; revised February 5, 2020; accepted February 17, 2020

Abstract—A control algorithm based on the required differential equations of deviations is proposed to solve the problem of tracing the desired trajectory by target position. The flight path is planned on the basis of the author's method of calculation of a flat trajectory consisting of oriented segments of straight lines connected by Euler spirals. The software and hardware complex of the automatic flight control system is presented. The operability of the control system in the presence of measurement noises and external disturbances is confirmed by the results of experiments to control the flight of a quadcopter indoors.

DOI: 10.3103/S8756699020030085

Keywords: *trajectory motion control, route flight, multirotor aircraft, structural synthesis method*

1. INTRODUCTION

In contrast to satellite systems, unmanned aerial vehicles (UAVs) make it possible to obtain images in cloudy conditions and provide higher spatial resolution. Today, it can be said that UAVs are successfully used in aerial photography and digital map generation. Multirotor vehicles are widely used in solving these problems due to controllability, reliability, and simplicity of construction. Promising applications of multirotor aircraft are cargo delivery, aero-geophysical works, and precision agriculture [1–4].

The application of UAVs is accompanied by formulating the problems of constructing the specified routes and the formation of control actions for traction motors taking into account the shape and size of the surveyed areas, terrain, external disturbances, and the dynamics of the used UAV.

When the robot moves in an unknown or complex environment, the motion trajectory is continuously planned and constructed [5, 6]. In this case, it is convenient to specify the trajectory by control points and further represent the trajectory in a parametric form [7]. If it is necessary to meet the requirements for the boundedness of the time derivatives of the target position coordinates or for the smoothness of the generated control actions, different methods of interpolation of the desired trajectory are used, e.g., the mathematical apparatus of the spline theory [8].

In this article, an algorithm for controlling the flight path of a multirotor UAV based on the developed method of structural synthesis is proposed. The target position is determined from the procedure of planning smooth motion trajectories defined by the sequence of turning points of the route using Cornu curves.

A prototype of the autonomous flight control system is created. The conducted experiments on the indoor quadcopter flight control showed that the proposed control system is operational.

*E-mail: kotov@idisys.iae.nsk.su

**E-mail: yan@idisys.iae.nsk.su

2. FORMULATION OF THE PROBLEM

The position of a multicopter aircraft in space is characterized by the coordinates x, y, z of the center of mass of the vehicle in a fixed Cartesian system and Euler's angles ψ, φ, θ , which determine the orientation of the coordinate system connected with the aircraft's axes in relation to the fixed system xyz [9].

Simplified dynamics equations describing motion in the specified coordinates are as follows:

$$\begin{cases} m\ddot{x} = (\sin \psi \cdot \sin \varphi + \cos \psi \cdot \cos \varphi \cdot \sin \theta)u_1; \\ m\ddot{y} = (-\cos \psi \cdot \sin \varphi + \sin \psi \cdot \cos \varphi \cdot \sin \theta)u_1; \\ m\ddot{z} = \cos \varphi \cdot \cos \theta \cdot u_1 - mg. \end{cases} \quad (1)$$

Here, m is the aircraft mass, g is the gravitational acceleration, and u_1 is the total traction force. Here and further, the dots above the variable signs indicate time derivatives.

Due to the fact that the built-in flight controller control system is used to adjust the required attitude of the aircraft, let us assume that the processes of changing the attitude angles are represented as independent second order differential equations [10]:

$$\begin{cases} T_\psi^2 \ddot{\psi} + 2d_\psi T_\psi \dot{\psi} + \psi = \psi_{\text{ref}}; \\ T_\varphi^2 \ddot{\varphi} + 2d_\varphi T_\varphi \dot{\varphi} + \varphi = \varphi_{\text{ref}}; \\ T_\theta^2 \ddot{\theta} + 2d_\theta T_\theta \dot{\theta} + \theta = \theta_{\text{ref}}. \end{cases} \quad (2)$$

The values of T and d coefficients were estimated experimentally.

The kinematic diagram determines the recalculation of the thrust force of the motors in the control torques on the corresponding axes of rotation. We assume that this recalculation is by default in the flight controller, and the control actions are the required thrust u_1 and the orientation given by the angles ψ_{ref} , φ_{ref} , and θ_{ref} .

We consider the control problem to be the trajectory control of the vehicle motion, namely, the placement of the center of mass of the vehicle into the target position with coordinates x_{ref} , y_{ref} , and z_{ref} and further following it. It is necessary to find such control actions u_1 , ψ_{ref} , φ_{ref} , and θ_{ref} which will ensure that the mismatch between the current and the target coordinates is reduced to zero. It is assumed that the coordinates of the center of mass and its orientation in space at every moment of time as well as the first derivatives of these values are known. It is also assumed that traction motors have sufficient power.

3. CONTROL ALGORITHM SYNTHESIS

The methods of interpolation of the required trajectory, such as splines [8] or the Cornu spiral theory [11], make it possible to determine the time derivatives of the target position coordinates on the constructed route. With this in mind, let us write down the required differential equations for the deviation of the center of mass of the vehicle from the target position x_{ref} , y_{ref} , z_{ref} :

$$\begin{cases} S_x = \dot{x} - \dot{x}_{\text{ref}} + k_x(x - x_{\text{ref}}); \\ S_y = \dot{y} - \dot{y}_{\text{ref}} + k_y(y - y_{\text{ref}}); \\ S_z = \dot{z} - \dot{z}_{\text{ref}} + k_z(z - z_{\text{ref}}). \end{cases} \quad (3)$$

Let us define the constrained motion to the desired trajectory ($S_x = 0$, $S_y = 0$, $S_z = 0$) by the equations

$$\dot{S}_x = -\alpha_x S_x, \quad \dot{S}_y = -\alpha_y S_y, \quad \dot{S}_z = -\alpha_z S_z. \quad (4)$$

Here, k_x, k_y, k_z and $\alpha_x, \alpha_y, \alpha_z$ are the positive constant coefficients that determine the time to reach the target position. The system synthesized based on Eqs. (3) and (4) is invariant with respect to the formulation of $x_{\text{ref}}, y_{\text{ref}}, z_{\text{ref}}$ and has astaticism.

By substituting (3) in (4), we obtain

$$\begin{cases} \ddot{x} - \ddot{x}_{\text{ref}} + k_x(\dot{x} - \dot{x}_{\text{ref}}) = -\alpha_x(\dot{x} - \dot{x}_{\text{ref}}) - \alpha_x k_x(x - x_{\text{ref}}); \\ \ddot{y} - \ddot{y}_{\text{ref}} + k_y(\dot{y} - \dot{y}_{\text{ref}}) = -\alpha_y(\dot{y} - \dot{y}_{\text{ref}}) - \alpha_y k_y(y - y_{\text{ref}}); \\ \ddot{z} - \ddot{z}_{\text{ref}} + k_z(\dot{z} - \dot{z}_{\text{ref}}) = -\alpha_z(\dot{z} - \dot{z}_{\text{ref}}) - \alpha_z k_z(z - z_{\text{ref}}). \end{cases} \quad (5)$$

Taking into account (1) and (2), let us exclude the highest derivatives of the controlled values in (5)

$$\begin{cases} (\sin \psi \cdot \sin \varphi + \cos \psi \cdot \cos \varphi \cdot \sin \theta)u_1 = -mA_x; \\ (-\cos \psi \cdot \sin \varphi + \sin \psi \cdot \cos \varphi \cdot \sin \theta)u_1 = -mA_y; \\ \cos \varphi \cdot \cos \theta u_1 = -mA_z, \end{cases} \quad (6)$$

where

$$\begin{cases} A_x = (\alpha_x + k_x)(\dot{x} - \dot{x}_{\text{ref}}) + \alpha_x k_x(x - x_{\text{ref}}) - \ddot{x}_{\text{ref}}; \\ A_y = (\alpha_y + k_y)(\dot{y} - \dot{y}_{\text{ref}}) + \alpha_y k_y(y - y_{\text{ref}}) - \ddot{y}_{\text{ref}}; \\ A_z = (\alpha_z + k_z)(\dot{z} - \dot{z}_{\text{ref}}) + \alpha_z k_z(z - z_{\text{ref}}) - \ddot{z}_{\text{ref}} - g. \end{cases} \quad (7)$$

Let us square the left and right parts of Eq. (6) and, having summed them up term by term, obtain the necessary total thrust of the engines

$$u_1 = m\sqrt{A_x^2 + A_y^2 + A_z^2} = mg_u. \quad (8)$$

System of equations (6) is not independent. This objectively reflects the fact that it is sufficient to control any two angles of ψ , φ , θ to control the course of the multirotor aircraft. When controlling, e.g., pitch and roll angles, the yaw angle ψ_{ref} can be selected from other considerations not related to changing the flight direction. According to the method described in [12], we find the necessary pitch and roll values of θ_{ref} and φ_{ref} from Eqs. (6):

$$\begin{cases} \theta_{\text{ref}} = \arctan (A_x \cos \psi_{\text{ref}} + A_y \sin \psi_{\text{ref}})/A_z; \\ \varphi_{\text{ref}} = \arctan (A_x \sin \psi_{\text{ref}} - A_y \cos \psi_{\text{ref}})/\sqrt{g_u^2 - (A_x \sin \psi_{\text{ref}} - A_y \cos \psi_{\text{ref}})^2}. \end{cases} \quad (9)$$

If the coefficients α and k are positive, the obtained control actions provide stable transients, and if $\alpha = k$, the processes will be monotonous. The solutions of these equations have the same form for different laws of changing control actions.

Input data for the control algorithm are noisy realizations of the coordinates of the center of mass of the aircraft. In addition, estimates of the first derivatives of these variables are required. Relations (1) and (2) can be used to estimate object state variables in the extended Kalman filter. Let us supplement the process model (1) and (2) with the observation model and write down the obtained nonlinear dynamic system in compact form

$$\begin{cases} \dot{X} = f(X, U) + W; \\ \dot{Z} = h(X) + V, \end{cases} \quad (10)$$

where $X = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \psi, \varphi, \theta, \dot{\psi}, \dot{\varphi}, \dot{\theta}]^\top$, $U = [u_1, \psi_{\text{ref}}, \varphi_{\text{ref}}, \theta_{\text{ref}}]^\top$, f is the nonlinear vector function corresponding to the right side of Eqs. (1) and (2), W is the input noise, and V is the measurement noise.

In the extended Kalman filter, nonlinear system (10) is linearized by decomposing the functions f and h into the Taylor series. Further estimation of the state vector X consists of successive extrapolation and correction stages [10].

Augmented reality (AR) coordinates of the marker fixed in the center of the vehicle mass, yaw, roll, and pitch angles, and angular velocities of the vehicle calculated by the onboard inertial system were used as measured components of the vector X . The resulting measurement vector is $Z = [x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}}, \psi_{\text{imu}}, \varphi_{\text{imu}}, \theta_{\text{imu}}, \dot{\psi}_{\text{imu}}, \dot{\varphi}_{\text{imu}}, \dot{\theta}_{\text{imu}}]^\top$. Unlike in [10, 13], here, the vector Z has no linear velocity components $\dot{x}_{\text{cam}}, \dot{y}_{\text{cam}}, \dot{z}_{\text{cam}}$. The use of these values, calculated by filtering the signals $x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}}$, causes a low-frequency oscillation in control system (9) caused by a delay that occurs during filtering.

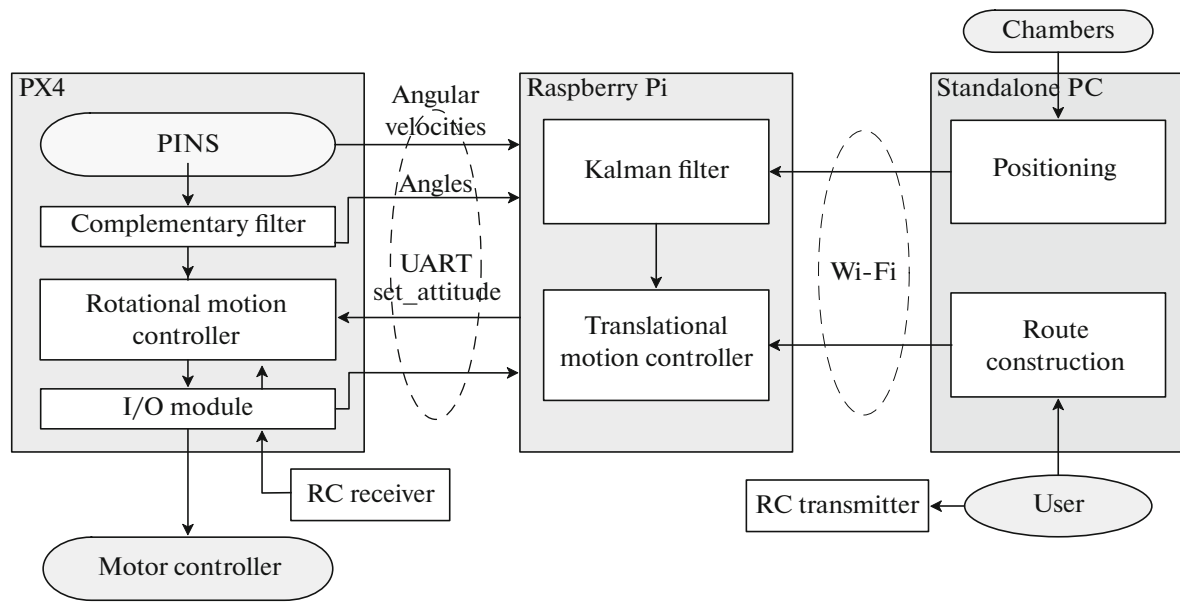


Fig. 1. Control system structural diagram.

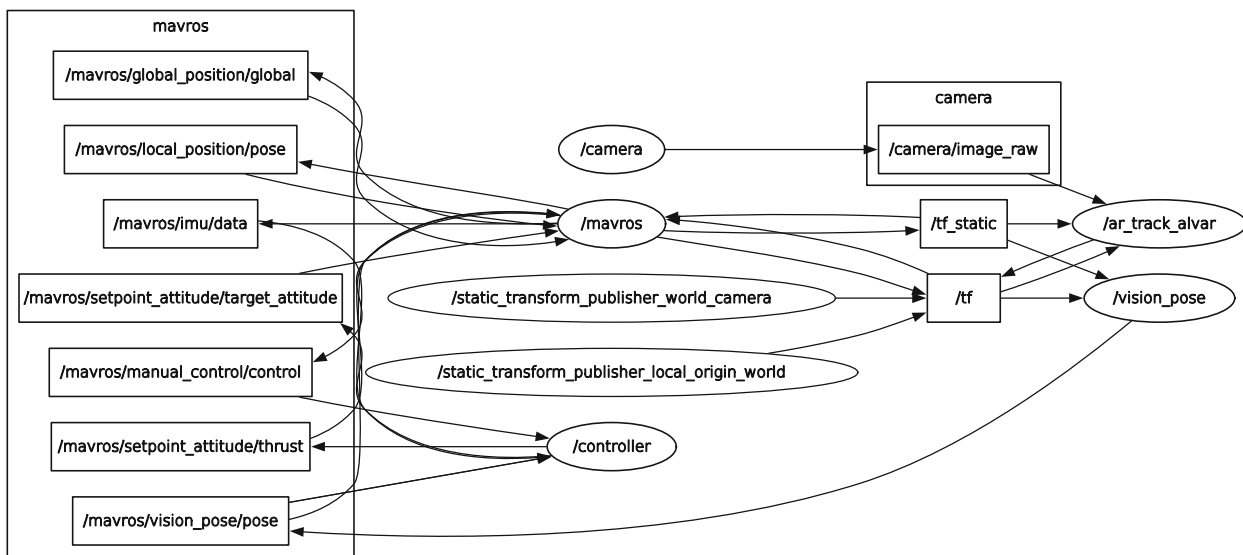


Fig. 2. Structural diagram of the software part of the control system.

4. CONTROL SYSTEM HARDWARE AND SOFTWARE DESCRIPTION

The developed control system has a two-level structure. The exchange of data between the controllers of the upper and lower levels is implemented via the mavlink protocol through a serial interface UART (Fig. 1). An autopilot Pixhawk with 32-bit STM32F427 Cortex M4 core processor and NuttX RTOS open-source operating system is used as a control controller of the lower level. The angular position of the vehicle is estimated in the module of the software complementary filter PX4 [14] using data from the integrated inertial sensor MPU 6000. The given orientation is set by the PX4 control module built in the software based on linear controllers. The autopilot operation modes are switched from the radio control (RC) console. In the automatic mode, the control command set_attitude containing the required values of Euler angles and total thrust of engines is submitted to the autopilot input. Takeoff and landing of the vehicle are performed in the manual mode by the control commands from the RC console. Absolute coordinates of the center of mass of the vehicle during indoor flights are determined using external video

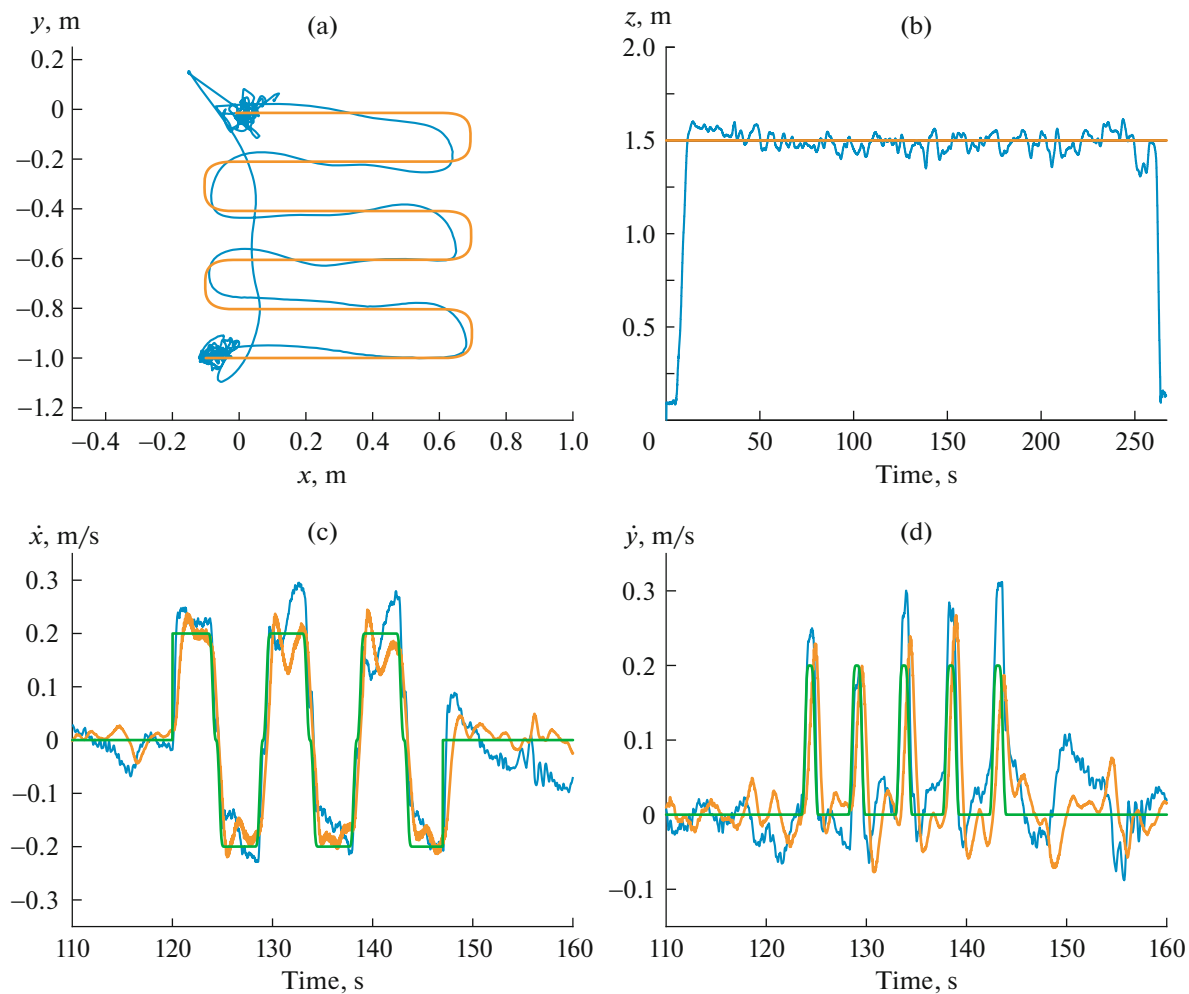


Fig. 3. Motion along the trajectory defined by the segments of the straight lines and Euler spirals: (a) the quadcopter trajectory in the plane (x, y) , (b) the height z above the surface, (c) the estimation of the linear velocity \dot{x} , and (d) the estimation of the linear velocity \dot{y} .

cameras and positioning system based on visual AR markers. The calculation of control actions taking into account the given flight path and current coordinates of the vehicle, integration, and filtering of these sensors is performed at the upper level; single board Raspberry Pi 3B microcomputer with 1.2 GHz 4-core Cortex-A53 processor (ARM v8) and 1 GB RAM. Ubuntu 16.04 LTS is used as the operating system.

The software of the created control system has a modular architecture and is built on the mechanism of interprocess communication called the Robot Operation System (ROS) [15]. In this article, we use ROS software packages designed to solve the problems of communication between controllers, detection of AR markers, and data storage. The modules of control and state estimation in the extended Kalman filter developed by the authors are written in the Python language using asynchronous processing based on ROS Timer and are integrated into the ROS.

The developed flight control system includes a set of independent programs that interact with each other by exchanging messages in the ROS. Its structure is shown in Fig. 2. The system includes software modules that perform the following functions:

- Receiving images from the camera (camera);
- Detection of AR markers (ar_track_alvar);
- Determination of the spatial position of the markers (vision_pose);
- Trajectory motion control and state vector estimation in the extended Kalman filter (controller);

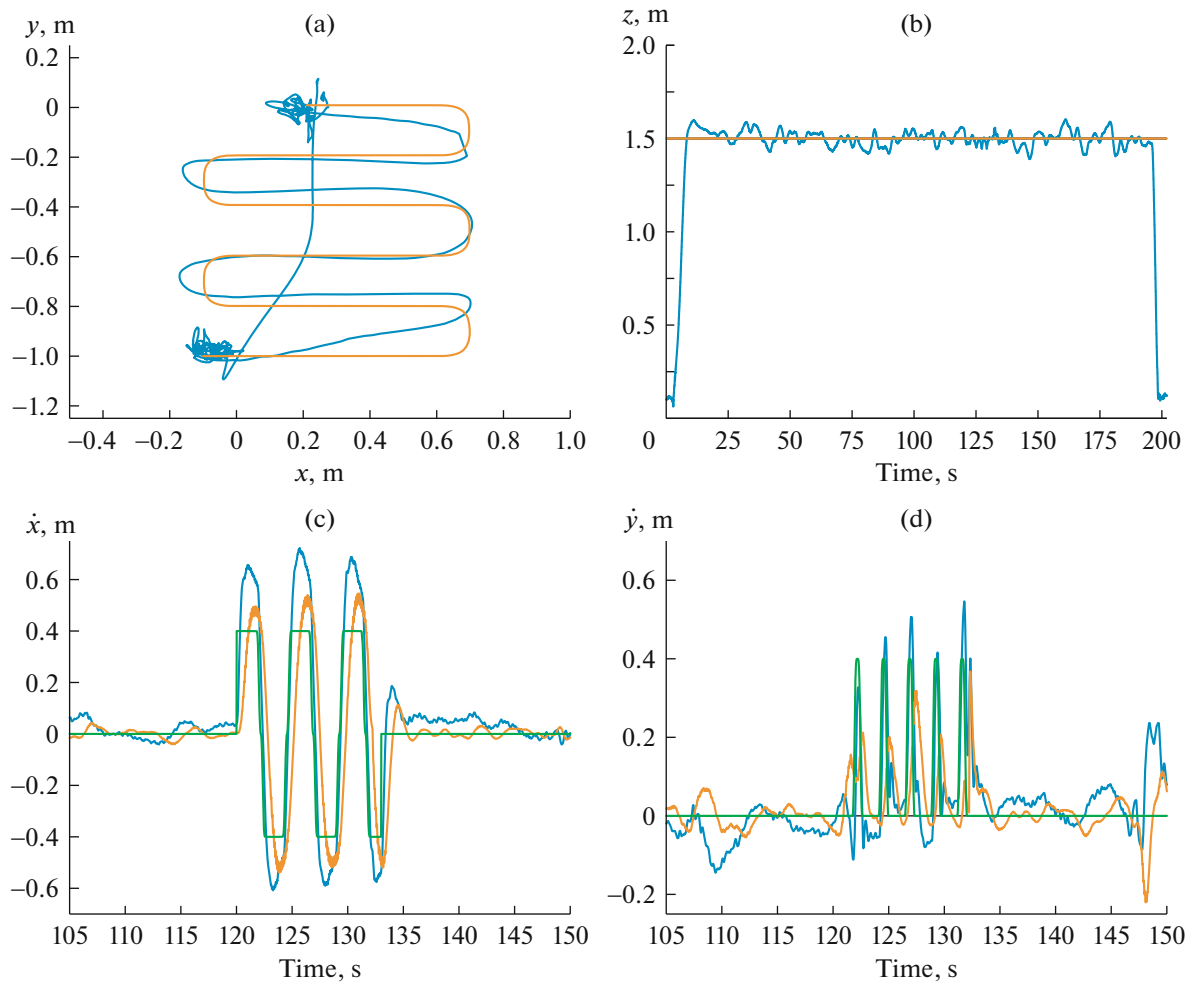


Fig. 4. Motion along the trajectory defined by the segments of the straight lines and Euler spirals (designations correspond to Fig. 3).

– Communication with the PX4 flight controller (mavros).

Absolute coordinates of the center of mass in space, attitude angles, and angular velocities relative to a fixed coordinate system are fed to the extended Kalman filter. The obtained coordinate estimations are used by the controller program, which calculates control commands for the required attitude angles and total thrust of the engines according to the developed trajectory motion algorithm. These commands are then sent via the mavros module to the PX4 flight controller for execution.

5. EXPERIMENTAL RESULTS

The method of trajectory planning using the Euler spiral for transition between segments of a straight line of the route defined by a sequence of turning points was implemented in the MATLAB/Simulink environment [10]. The generated program trajectory is fed to the input of control algorithm (9). When constructing the trajectory, the following parameter values were used: the linear velocity of the target $V = 0.2; 0.4$ m/s and the maximum normal g-force $n_{enable} = 0.08; 0.3$.

Aircraft motion Eqs. (1) and (2) have the following parameters: $m = 0.4$ kg, $T_\psi = T_\varphi = T_\theta = 0.17$ s, and $d_\psi = d_\varphi = d_\theta = 0.35$, and for the control system we have $k_x = \alpha_x = 4.0$, $k_y = \alpha_y = 4.0$, and $k_z = \alpha_z = 4.0$.

Control commands `setpoint_attitude` were fed with a sampling frequency of 0.01 s. Data on the attitude of the vehicle calculated by its navigation system as well as coordinates of the center of mass of the vehicle from the external video system came at intervals of 0.02 and 0.05 s, respectively.

Before the motion started, the vehicle executed a command to hold a stationary target position until the time $t = 120$ s. The time $t = 0$ s corresponds to the startup of the controller control program. The calculated value of the standard deviation from the average variables x, y, z was 0.03 m. The calculation was performed in a time interval from 30 to 115 s; the values exceeding the originally calculated value by 3.5 in absolute value were discarded.

Figure 3 shows the results for the case of motion along the trajectory with the parameters $V = 0.2$ m/s, $n_{\text{enable}} = 0.08$. The deviation from the trajectory during the motion was 0.05 m.

The estimation of the translational velocity components \dot{x}, \dot{y} , obtained in the Kalman's filter was compared with the estimation of linear velocities obtained by filtering signals x, y . The difference in estimation was 0.1 m/s (Figs. 3c and 3d).

Increase of the velocity of motion along the programmed trajectory up to $V = 0.4$ m/s and g-force up to $n_{\text{enable}} = 0.3$ leads to the decrease in accuracy of maintaining the given position during maneuvers (Fig. 4).

6. CONCLUSIONS

In this article, the algorithm for trajectory motion control of a multirotor vehicle in the problem of target position tracking was proposed. The hardware and software complex of the automatic flight control system was developed. The efficiency of the previously proposed route planning method using Cornu curves, which provides a smooth change of g-force on turns, was experimentally confirmed. The achieved indoor positioning accuracy of the quadcopter relative to the desired trajectory was 0.05–0.1 m.

FUNDING

This work was supported by the Ministry of Science and Higher Education of the Russian Federation, project no. AAAA-A17-117060610006-6.

REFERENCES

1. M. Kuckelhaus, D. Niezgoda, and S. Endriß, *Unmanned Aerial Vehicles in Logistics* (DHL, Troisdorf, 2014).
2. V. Duggal, M. Sukhwani, K. Bipin, G. S. Reddy, and K. M. Krishna, "Plantation monitoring and yield estimation using autonomous quadcopter for precision agriculture," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Stockholm, Sweden, 2016*, pp. 5121–5127. <https://doi.org/10.1109/ICRA.2016.7487716>
3. P. Ćwiąkała, R. Kocierz, E. Puniach, M. Nędzka, K. Mamczarz, W. Niewiem, and P. Wiącek, "Assessment of the possibility of using unmanned aerial vehicles (uavs) for the documentation of hiking trails in alpine areas," *Sensors* **18**, 81 (2017). <https://doi.org/10.3390/s18010081>
4. Ch. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: a review," *Precis. Agric.* **13**, 693–712 (2012). <https://doi.org/10.1007/s11119-012-9274-5>
5. O. Brock and O. Khatib, "Real-time replanning in high-dimensional configuration spaces using sets of homotopic paths," in *Proc Int. Conf. on Robotics and Automation (ICRA), San Francisco, 2000*, pp. 550–555. <https://doi.org/10.1109/ROBOT.2000.844111>
6. J. S. Pershina, S. Ya. Kazdorf, and A. V. Lopota, "Methods of mobile robot visual navigation and environment mapping," *Optoelectron., Instrum. Data Process.* **55**, 181–188 (2019). <https://doi.org/10.3103/S8756699019020109>
7. S. Thrun, M. Montemerlo, H. Dahlkamp, et al., "Stanley: The robot that won the DARPA Grand Challenge," *J. Field Rob.* **23**, 661–692 (2006). <https://doi.org/10.1002/rob.20147>
8. J. Connors and G. Elkaim, "Analysis of a spline based, obstacle avoiding path planning algorithm," in *Proc. IEEE 65th Vehicular Technology Conf. (VTC2007-Spring), Dublin, 2007*, pp. 2565–2569. <https://doi.org/10.1109/VETECS.2007.528>
9. J. Kim, M.-S. Kang, and S. Park, "Accurate modeling and robust hovering control for a quad-rotor vtol aircraft," *J. Intell. Rob. Syst.* **57**, 9–26 (2010). <https://doi.org/10.1007/s10846-009-9369-z>
10. S. A. Belokon', Yu. N. Zolotukhin, K. Yu. Kotov, A. S. Mal'tsev, A. A. Nesterov, V. Ya. Pivkin, M. A. Sobolev, M. N. Filippov, and A. P. Yan, "Using the Kalman filter in the quadrotor vehicle trajectory tracking system," *Optoelectron., Instrum. Data Process.* **49**, 536–545 (2013). <https://doi.org/10.3103/S8756699013060022>

11. S. A. Belokon', Yu. N. Zolotukhin, and A. A. Nesterov, "Aircraft path planning with the use of smooth trajectories," *Optoelectron., Instrum. Data Process.* **53**, 1–8 (2017).
<https://doi.org/10.3103/S8756699017010010>
12. S. A. Belokon', Yu. N. Zolotukhin, A. S. Mal'tsev, A. A. Nesterov, M. N. Filippov, and A. P. Yan, "Control of flight parameters of a quadrotor vehicle moving over a given trajectory," *Optoelectron., Instrum. Data Process.* **48**, 454–461 (2012). <https://doi.org/10.3103/S8756699012050044>
13. A. S. Dimova, K. Yu. Kotov, A. S. Mal'tsev, A. A. Nesterov, and M. N. Filippov, "Quadrotor control in payload transportation on suspension," *Optoelectron., Instrum. Data Process.* **54**, 520–524 (2018).
<https://doi.org/10.3103/S8756699018050151>
14. L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Seattle, 2015*, pp. 6235–6240. <https://doi.org/10.1109/ICRA.2015.7140074>
15. M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software 3, 2009*, p. 5.

Translated by O. Pismenov