
**SYSTEMS OF CONTROL AND INFORMATIONAL
MEASUREMENT**

**Motion Control System for Quadcopter Based
on Cascade Kalman Filters**

F. P. Gerasimov¹, Yu. N. Zolotukhin¹, K. Yu. Kotov^{1,2*}, A. S. Mal'tsev¹, and A. P. Yan¹

¹*Institute of Automation and Electrometry, Novosibirsk, 630090 Russia*

²*Synchrotron Radiation Facility—Siberian Circular Photon Source SKIF
Boreskov Institute of Catalysis of Siberian Branch of the Russian Academy of Sciences,
Koltsovo, Novosibirsk oblast, 630559 Russia*

Received June 27, 2022; revised June 29, 2022; accepted July 11, 2022

Abstract—To solve the problem of estimating the orientation and position of an aircraft, we propose a system based on a series of Kalman filters: a linear Kalman filter in the attitude stabilization loop and an extended Kalman filter in the subsystem of maintaining a given location. The efficiency of the developed control system is confirmed by experimental results on the indoors flight control of a quadcopter.

DOI: 10.3103/S8756699022040069

Keywords: *automatic flight control system, cascaded Kalman filters*

1. INTRODUCTION

When creating motion control systems for aircraft, manipulators, autonomous mobile robots, or human positioning systems, it becomes necessary to solve the problem of estimating the state vector of an object containing spatial coordinates and movement velocities. The systems for determining attitude, position and velocity, described in works [1–7], include inertial measurement modules (IMMs) and algorithms for data processing and integration.

Currently, IMMs that consist of an assembly of micromechanical sensors—accelerometers, gyrosensors and magnetometers, are ubiquitous. Navigation systems with IMMs are complemented by other types of sensors—satellite signal receivers, optical and ultrasonic sensors.

In a number of cases algorithms for estimating attitude are based on simple state observers—complementary filters. An algorithm based on a nonlinear complementary filter using data from inertial sensors and a magnetometer was proposed in [1]. The work [2] considers an adaptive complementary filter based on quaternions. In [3], an algorithm based on the gradient descent method is described; and in [4] an alternative extended complementary filter is developed that does not require minimization procedures.

Estimation of aircraft location requires integration of data on the attitude and position of the object. Methods of optimal linear filtering are used to solve this problem. The linear Kalman filter is suggested for estimating the coordinates of a nonlinear double pendulum system [5]. The gradient descent method for estimating the measured attitude quaternion and the linear Kalman filter are used to estimate the attitude of the quadcopter in [6]. In [7], an algorithm was proposed for integrating data from an IMM and an optical flow sensor using an extended Kalman filter in the control system of a multirotor aircraft.

A kinematic or simplified dynamic (with constant accelerations) model of the vehicle motion is usually used for constructing a coordinate estimation system based on the Kalman filter. Often, model inputs corresponding to real control actions applied to the object are not considered.

The purpose of this work is to create a system for estimating the attitude and position of an aircraft based on series-connected or cascade Kalman filters [8]. It is proposed to use a linear Kalman filter

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

for a fast attitude stabilization loop and an extended Kalman filter for the subsystem of maintaining a given position. The efficiency of the developed control system is confirmed by experimental results on the indoor flight control of a quadcopter.

2. PROBLEM SETUP

The spatial position of the quadcopter is described by the coordinates of the center of mass of the device x , y , z in the inertial frame of reference and by the angles of rotation around the axes x_b , y_b , z_b of the frame of reference rigidly bound to the device with the origin at the center of gravity of the quadcopter. We accept the following designations: ϕ is the angle of rotation around x_b axis (roll), θ is the angle of rotation around axis y_b (pitch), and ψ is the angle of rotation around axes z_b (yaw). A simplified mathematical model that characterizes the movement of a quadcopter in these coordinates is given by the equations [9]

$$\begin{cases} m\ddot{x} = (\sin \psi \cdot \sin \phi + \cos \psi \cdot \cos \phi \cdot \sin \theta)u_1; \\ m\ddot{y} = (-\cos \psi \cdot \sin \phi + \sin \psi \cdot \cos \phi \cdot \sin \theta)u_1; \\ m\ddot{z} = \cos \phi \cdot \cos \theta \dot{u}_1 - mg, \end{cases} \quad (1)$$

$$\begin{cases} I_{xx}\ddot{\phi} = u_2 - (I_{zz} - I_{yy})\dot{\theta}\dot{\psi}; \\ I_{yy}\ddot{\theta} = u_3 - (I_{xx} - I_{zz})\dot{\phi}\dot{\psi}; \\ I_{zz}\ddot{\psi} = u_4, \end{cases} \quad (2)$$

where m is the quadcopter mass; g is the acceleration of gravity; I_{xx} , I_{yy} , I_{zz} are moments of inertia about the principal axes of the quadcopter; u_1 is the thrust force along the z_b axis; and u_2 , u_3 , u_4 are torque moments. Dots over variables denote time derivatives.

Equations (1) and (2) can be represented in a compact form:

$$\dot{q} = f(q(t), u(t)), \quad (3)$$

where

$$q = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T \quad (4)$$

is the system state vector; $f(q, u)$ is a vector function defined by the object model; and u is the vector of control actions.

Figure 1 shows forces and moments acting on the quadcopter. The aircraft frame has a diagonal length of $2l_0$ m and is rotated at an angle $\gamma = \pi/4$ rad about the z_b axis of the body-fixed reference system. Each rotor generates aerodynamic thrust F_i directed upwards and moment M_i directed opposite to the rotation of the propeller i . The total thrust is given by the ratio

$$u_1 = F_1 + F_2 + F_3 + F_4. \quad (5)$$

In the quadcopter coordinate system, the roll and pitch torques are calculated using the right hand rule. Moments of force over x_b and y_b axes are, respectively,

$$u_2 = l(F_1 + F_2 - F_3 - F_4), \quad (6)$$

$$u_3 = l(-F_1 + F_2 + F_3 - F_4), \quad (7)$$

where $l = l_0 \sin \gamma$.

For the z_b axis, the thrust of the motors does not create a moment of force. However, the rotation of the propellers generates an aerodynamic moment directed against the rotation of the blades of each of the four propellers:

$$u_4 = M_1 - M_2 + M_3 - M_4.$$

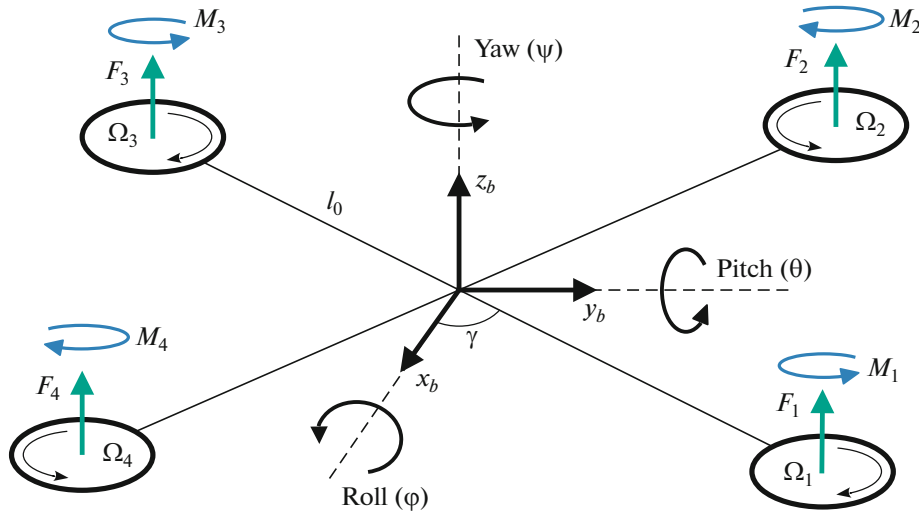


Fig. 1. Coordinate systems, forces and moments acting on the quadcopter.

The transition from the forces created by each rotor to the thrust and moments of the entire system is written in matrix form:

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ l & l & -l & -l \\ l & -l & -l & l \\ \lambda & -\lambda & \lambda & -\lambda \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} = M \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix},$$

where λ is the coefficient of proportionality between aerodynamic moments along the z_b axis and propeller thrusts. Thus, the thrust forces that the control action creates on each engine are calculated as

$$\begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} = M^{-1} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}.$$

We consider the task of control to be the trajectory control of the movement of the aircraft, namely, bringing the center of mass of the craft to the target position with coordinates x_{ref} , y_{ref} , z_{ref} and following it. Papers [10, 11] present a method for synthesizing control actions $u_1 - u_4$, which ensure that the mismatch between the current and target coordinates is reduced to zero.

The implementation of control algorithms requires estimates of the coordinates of the center of mass of the aircraft and its attitude angles at each moment of time, as well as the first derivatives of these variables.

This work aims to create a system for estimating the above state variables of the aircraft using a mathematical motion model and motion control algorithms.

3. ALGORITHMS AND ARCHITECTURE OF THE ATTITUDE AND POSITION ESTIMATION SYSTEM

The motion model (1), (2) prescribes a natural division of the problem of estimating the coordinates into two stages.

First of all, it is necessary to estimate the attitude angles of the aircraft from the noisy IMM data.

Values of torques that implement monotonous and stable processes of establishing the required values of angles ϕ_{ref} , θ_{ref} , ψ_{ref} , are presented in [11]:

$$\begin{cases} u_2 = I_{xx}[-(\alpha_\phi + k_\phi)\dot{\phi} - \alpha_\phi k_\phi(\phi - \phi_{\text{ref}})]; \\ u_3 = I_{yy}[-(\alpha_\theta + k_\theta)\dot{\theta} - \alpha_\theta k_\theta(\theta - \theta_{\text{ref}})]; \\ u_4 = I_{zz}[-(\alpha_\psi + k_\psi)\dot{\psi} - \alpha_\psi k_\psi(\psi - \psi_{\text{ref}})], \end{cases} \quad (8)$$

where α and k and constant parameters that determine the type and duration of the transition process according to the controlled values.

At the next stage, it is necessary to estimate the coordinates of the aircraft location and the linear speeds of movement according to the data of the external coordinate system and the attitude angles of the aircraft. The values of angles ϕ_{ref} , θ_{ref} , required for the quadcopter to steadily reach the target position x_{ref} , y_{ref} , z_{ref} , have the following form:

$$\begin{cases} \phi_{\text{ref}} = \arctan(H_{xx}/H_{zz}); \\ \theta_{\text{ref}} = \arctan(-H_{yy}/\sqrt{H_{xx}^2 + H_{zz}^2}), \end{cases} \quad (9)$$

where

$$\begin{cases} H_{xx} = -(\alpha_x + k_x)\dot{x} - \alpha_x k_x(x - x_{\text{ref}}); \\ H_{yy} = -(\alpha_y + k_y)\dot{y} - \alpha_y k_y(y - y_{\text{ref}}); \\ H_{zz} = -(\alpha_z + k_z)\dot{z} - \alpha_z k_z(z - z_{\text{ref}}) + g. \end{cases} \quad (10)$$

To control a quadcopter, it is enough to control any two angles, so we assume $\psi_{\text{ref}} = 0$. The thrust force is given as

$$u_1 = m\sqrt{H_{xx}^2 + H_{yy}^2 + H_{zz}^2}. \quad (11)$$

3.1. Attitude Estimation

The main source of noise in the received readings of linear accelerations and angular velocities from the IMMs are vibrations caused by the operation of propelling motors. To eliminate the noise component, low-pass filters are used, for example, Butterworth filters or the simplest filters with a first-order aperiodic link transfer function:

$$\frac{y^f}{y} = \frac{1}{\tau p + 1}, \quad (12)$$

where τ is a parameter that defines the filter cutoff frequency, y and y^f are the original signal and the filter output signal, and p is the differentiation operator.

To take into account the phase shift and amplitude distortions introduced by the filter, we supplement the model (2) with signals of angular velocities along the x, y, z axes and attitude angles (calculated using linear accelerations) at the input and output of the filter (12).

For example, for the x_b axis we have

$$\begin{cases} \dot{w}_x = u_2/I_{xx}; \\ \dot{w}_x^f + k_f w_x^f = k_f w_x; \\ \dot{\phi} = w_x; \\ \dot{\phi}^f + k_f \phi^f = k_f \phi, \end{cases} \quad (13)$$

where $k_f = 1/\tau$ is the filter coefficient.

Equations (13) do not take into account the cross-link component between the axes y and z , which occurs when the aircraft rotates along the yaw angle. To simplify the obtained relations, we assume that the control is carried out by an independent change in the roll and pitch angles.

For the angular velocities w_y , w_z and angles ψ , θ , ψ , Eqs. (13) are the same. Control actions $u_2 - u_4$ are defined by relations (8).

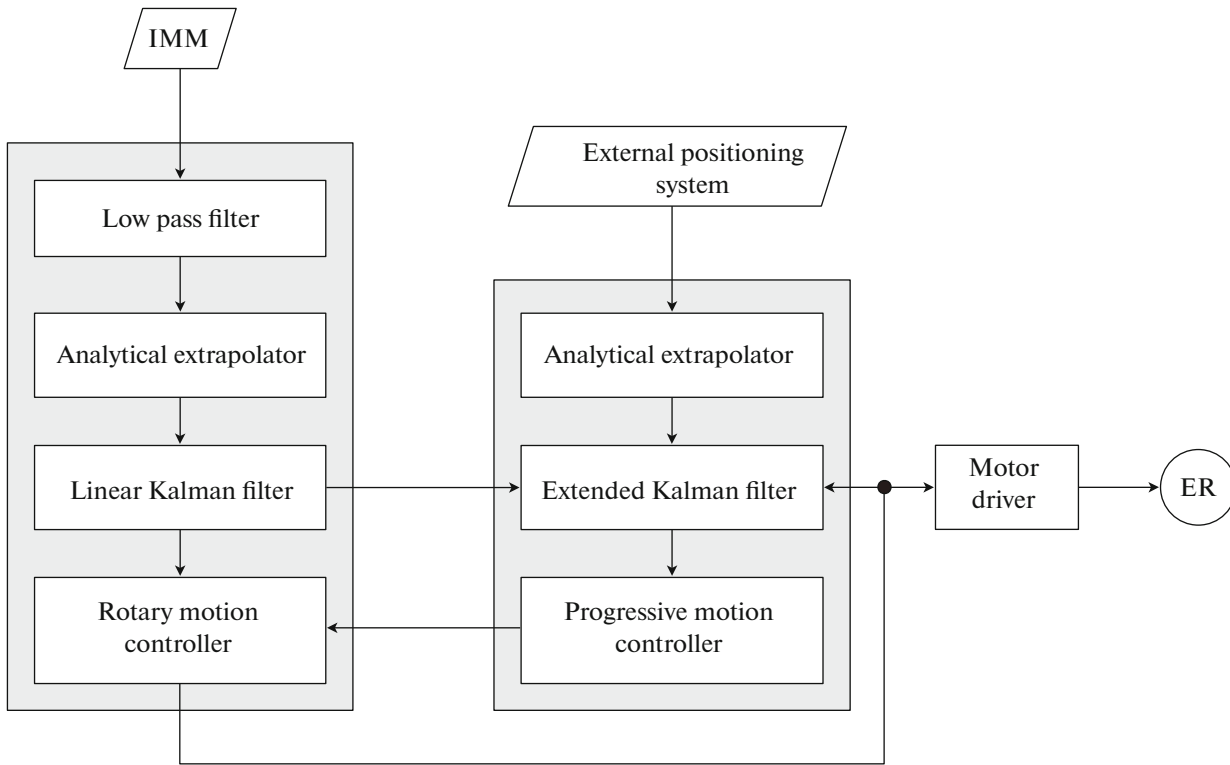


Fig. 2. Quadcopter control system.

In addition to noise, transport lags inevitably arise in the system due to the finite speed of the data transmission channels, different operating frequencies of the modules, and sampling of the measured continuous quantities. The total lag time in a discrete system is related to the sampling period Δt by the relation

$$T = N \Delta t. \tag{14}$$

One of the ways to compensate for the transport lag T is to use an analytical extrapolator, which makes it possible to estimate the state vector q for the current time t , using the mathematical model of the control object [12]. The estimation is written as

$$\hat{q}^k = q_d^k + \sum_{i=k}^{k+N} f(q_m^i, u^{i-N}) \Delta t, \tag{15}$$

where \hat{q}^k is the estimation of the state vector of the system at a discrete moment of time $k\Delta t$; q_d^k is the measurement of the state vector at the current time; and q_m^i is the state of model (4), initialized by the value $q_m^k = q_d^k$. Using the extrapolator requires storing N of the latest control values u^i received by the object.

Lag compensation using an extrapolator leads to an increase in noise due to the fact that the integral over N samples is added to the original vector, which leads to error accumulation. To eliminate deviations and outliers relative to the state vector, it is suggested to apply a linear Kalman filter.

For the x axis the estimated state vector has the form

$$\hat{q}_x^k = [w_x^k, w_x^{fk}, \phi^k, \phi^{fk}]^T. \tag{16}$$

Kalman filtering includes prediction steps

$$\hat{q}_x^{k+1} = F^k \hat{q}_x^k + B^k u^k \tag{17}$$

and correction steps:

$$P^{k+1} = F^k P^k (F^k)^T + Q^k, \tag{18}$$

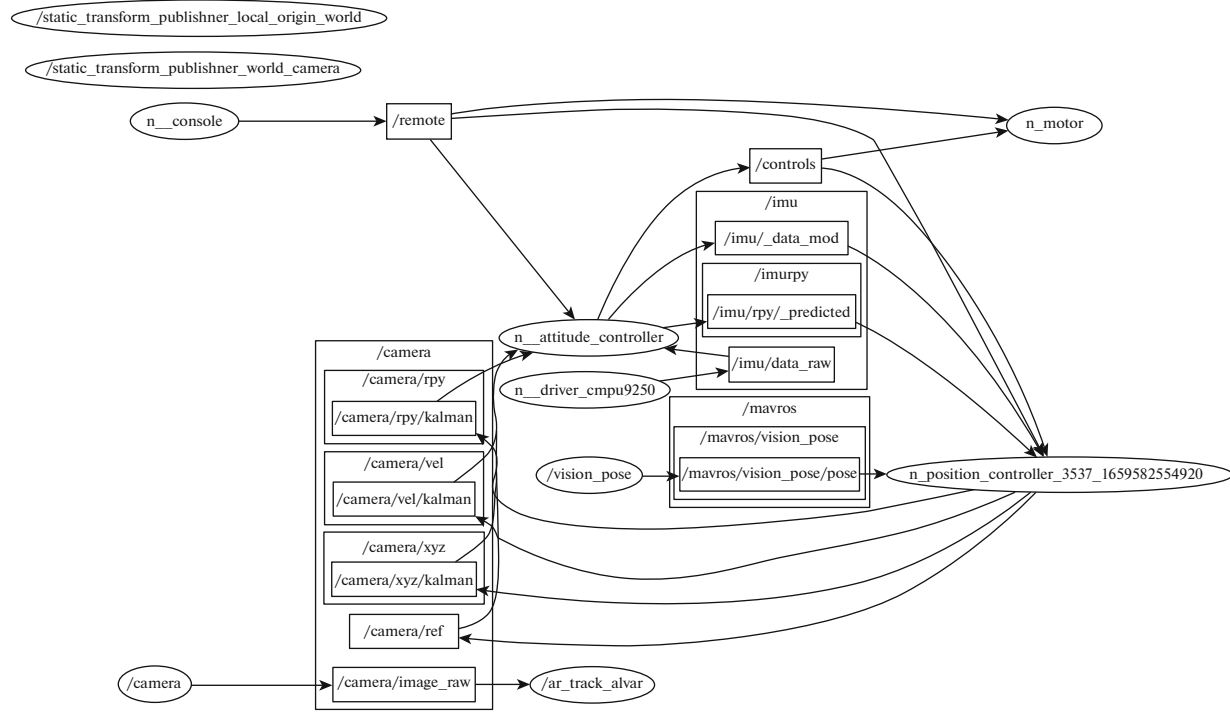


Fig. 3. Block diagram of the software of the control system in the ROS environment.

$$K^k = P^k / (P^k + R^k), \tag{19}$$

$$\hat{q}_x^{k+1} = \hat{q}_x^k + K^k [z_x^k - H \hat{q}_x^k], \tag{20}$$

where F^k is the model motion matrix (13); B^k is the control function matrix; H^k is the measurement model matrix; P^k , R^k , and Q^k are covariance state matrices, measurement noise, and model error, respectively.

3.2. Coordinate Estimation

In addition to attitude angles and angular velocities, the input data for the control algorithm (8)–(11) are noisy realizations of the coordinates of the center of mass of the aircraft, obtained from an external positioning system. Additionally, the estimates of the first derivatives of these variables are required. Relations (1) and (2) can be used to evaluate object state variables in the extended Kalman filter.

Similar to the lower layer, the transport time lags associated with the transmission of signals are eliminated by the analytical extrapolator (15).

Since the model (3) is nonlinear, in order to smooth the extrapolated state vector, it is proposed to apply the extended Kalman filter, in which the complete mathematical model of the aircraft is linearized near the operating point (\hat{q}_{ekf}^k, u^k) using the Taylor series expansion:

$$\hat{q}_{ekf}^{k+1} \approx f(\hat{q}_{ekf}^k, u^k) + F^k [\hat{q}^k - \hat{q}_{ekf}^k] + \omega(k), \tag{21}$$

$$F^k = \left. \frac{\partial f(q^k, u^k)}{\partial q^k} \right|_{q^k = \hat{q}_{ekf}^k}, \tag{22}$$

where q is the total state vector (4) and $\omega(k)$ is a random process describing modeling errors. The prediction step of the extended Kalman filter differs from the linear case (17) and is written as

$$\hat{q}_{ekf}^{k+1} = f(\hat{q}_{ekf}^k, u^k). \tag{23}$$

The correction stage is similar to the linear Kalman filter and is calculated using formulas (18)–(20).

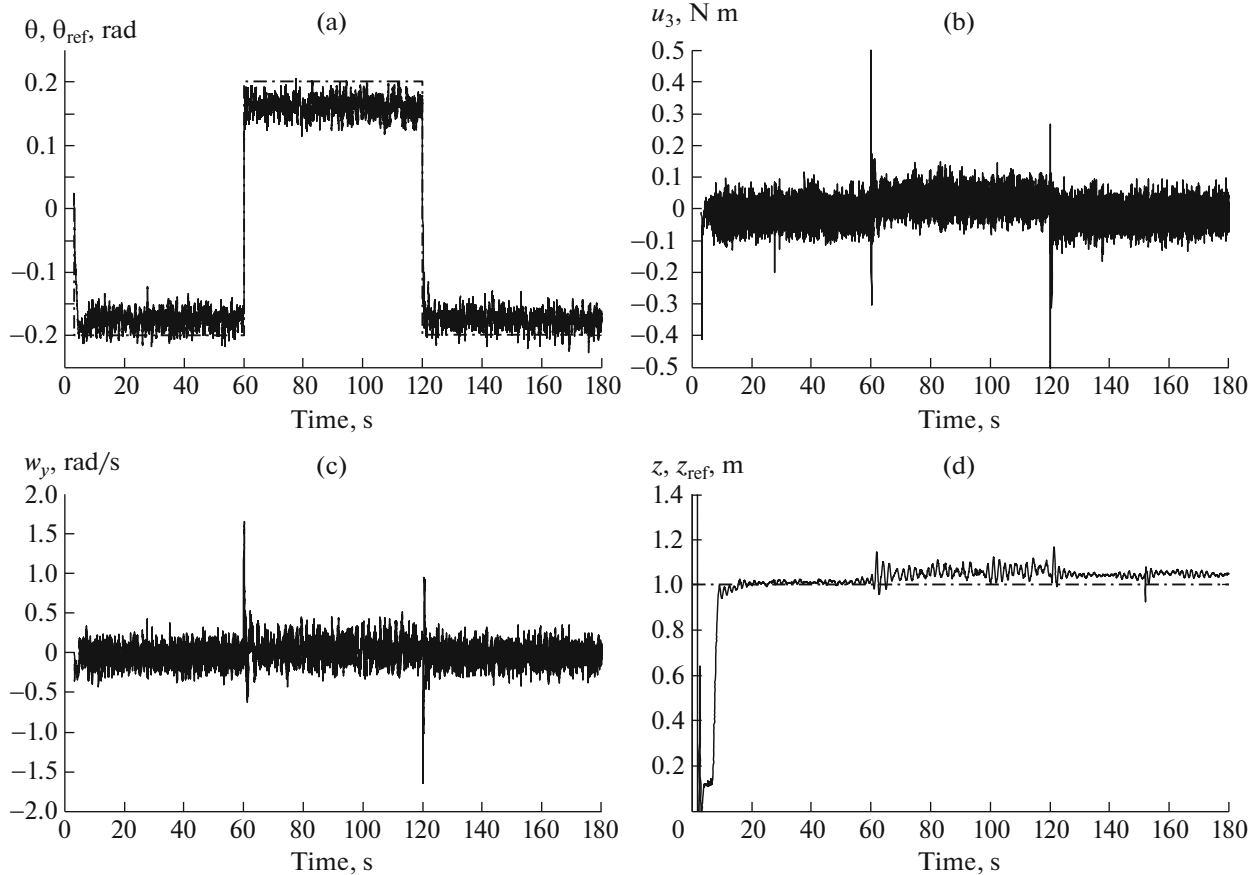


Fig. 4. Changing the pitch angle of the aircraft in stabilization mode: (a) estimate θ and required value of the pitch angle θ_{ref} ; (b) control moment u_3 ; (c) estimate of the angular velocity w_y ; (d) current height z (dash-dotted lines mark θ_{ref} , x_{ref}).

The input of the extended Kalman filter receives estimates of angular velocities and attitude angles obtained in the linear Kalman filter at the attitude estimation stage. The resulting measurement vector has the form

$$z^k = [x^k, y^k, z^k, \phi^k, \theta^k, \psi^k, w_x^k, w_y^k, w_z^k]^T. \quad (24)$$

The control calculated at the first stage is fed to the extended Kalman filter and to the software module of the motors, where, taking into account the transition matrix and the power characteristics of the motors, a pulse-width modulated signal is generated that determines the speed of rotation of the rotors connected to the electronic speed controllers (Fig. 2).

4. DESCRIPTION OF SOFTWARE AND HARDWARE OF THE CONTROL SYSTEM

Calculation of control actions and current coordinates of the aircraft, integration and filtering of sensor data, and generation of control signals to motors are performed on a Raspberry Pi 3B single-board microcomputer with a 4-core Cortex-A53 (ARM v8) 1.2 GHz processor and 1 GB RAM. The operating system is Ubuntu 20.04 LTS.

The software of the created control system has a modular architecture and is built on the ROS (Robot Operating System) interprocess communication mechanism [13].

In this work, ROS software packages are used, designed to solve the problems of obtaining IMM data, recognizing AR markers, and saving data (Fig. 3). The developed modules for controlling and estimating the state in the linear and extended Kalman filters are written in C++/Python using asynchronous processing based on ROS Timer and integrated into the ROS system. Takeoff and landing of the aircraft are carried out in manual mode according to the operator's commands, which are

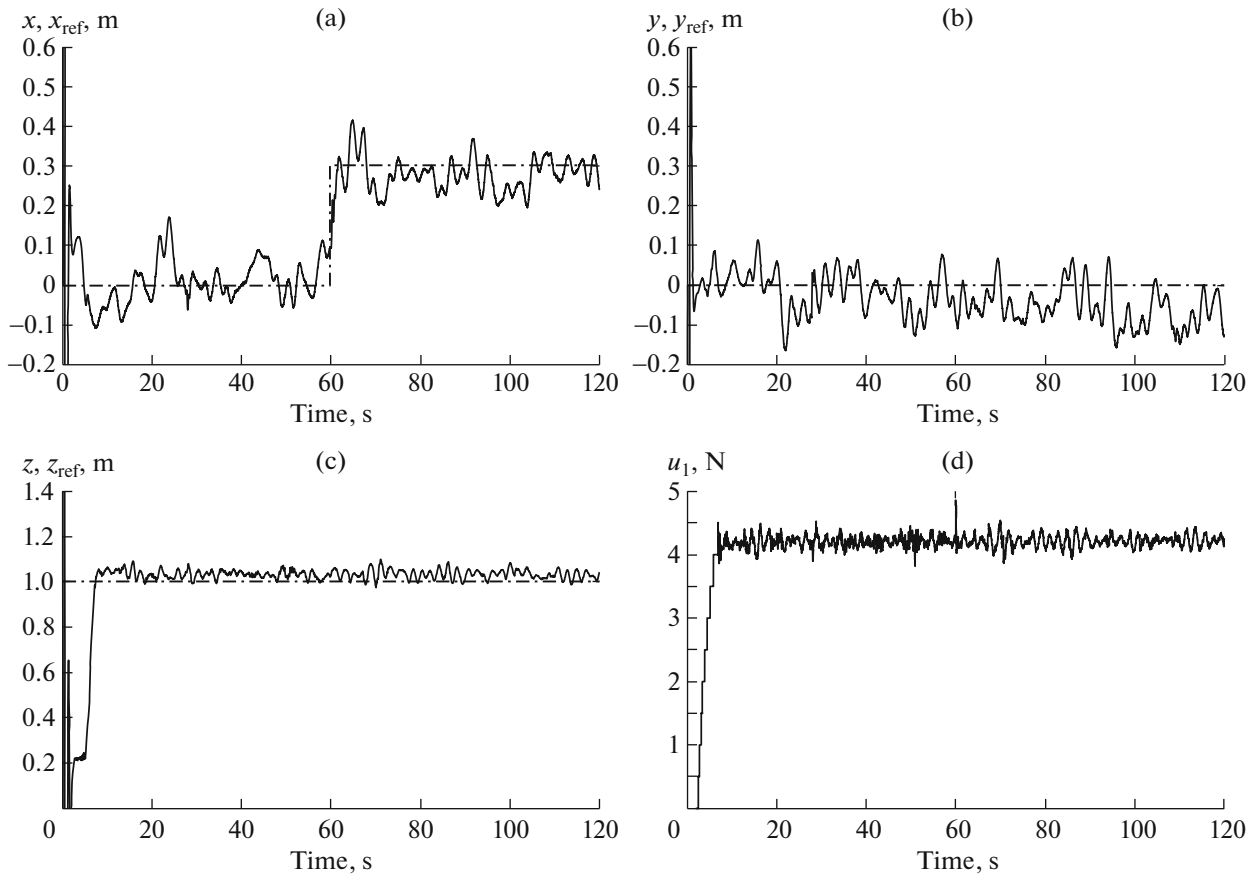


Fig. 5. Changing the aircraft coordinates in location maintenance mode: (a) coordinates x and x_{ref} ; (b) coordinates y and y_{ref} ; (c) coordinates z and z_{ref} ; (d) thrust force u_1 (dash-dotted lines mark x_{ref} , y_{ref} , z_{ref}).

processed in the console module. The absolute coordinates of the center of mass of the aircraft during indoor flights are determined using an external video camera and a localization system based on visual AR markers camera, `ar_track_alvar` and `vision_pose` modules.

The developed control system has a two-level structure.

The estimation of the angular position and the establishment of the given attitude of the aircraft are performed in the `attitude_controller` module using the data of the integrated inertial module InvenSense MPU-9250.

Estimation of the position of the aircraft and the establishment of the specified coordinates are performed in the `position_controller` module using data from the lower-level module and data from an external video positioning system.

In the motor software module, a linear dependence of the width of the pulse-width modulation signal on the thrust force of force F_i i th motor is implemented. This dependence has been estimated experimentally using a tensometric balance and, in the working area of the thrust force, it is represented with sufficient accuracy by a linear dependence.

5. EXPERIMENTAL RESULTS

To evaluate the effectiveness of the proposed control system a quadcopter prototype was assembled based on an X-shaped frame with a diagonal of 250 mm in the laboratory environment. The aircraft has the following mass-inertial characteristics: $m = 0.4$ kg, $J_x = 7.5 \times 10^{-3}$ kg m², $J_y = 7.5 \times 10^{-3}$ kg m², $J_z = 14.2 \times 10^{-3}$ kg m².

The control signals were supplied to the motors by the software module `motor_driver` at a frequency of 400 Hz. The frequency of the `attitude_controller` and `position_controller` modules is set to 200 and

100 Hz, respectively. Data on the attitude of the aircraft, calculated by its navigation system, as well as the coordinates of its center of mass from an external video system, were received at a frequency of 270 and 20 Hz, respectively.

Test bench checks included a series of flights in the stabilization mode or maintaining the given angles and flights in the mode of maintaining the given coordinates x_{ref} , y_{ref} . Maintenance of the altitude z_{ref} was performed in all modes.

The quality of transition processes was evaluated with a stepwise change in the controlled variable.

The following parameters of the control algorithm were set: $k_{\theta} = \alpha_{\theta} = 16$; $k_{\psi} = \alpha_{\psi} = 16$; $k_{\phi} = \alpha_{\phi} = 16$ and $k_x = \alpha_x = 4$; $k_y = \alpha_y = 4$; $k_z = \alpha_z = 4$.

In the stabilization mode over angles θ and ψ (the specified heading angle $\phi_{\text{ref}} = 0$ in all cases) the transition characteristics obtained for these parameters are monotonous without overshoot (Fig. 4). Settling time for specified pitch and roll angles θ_{ref} , ψ_{ref} after switching at times $t = 60$ s and $t = 120$ s does not exceed 0.2 s.

Figure 5 presents the results of maintaining the specified coordinates x_{ref} , y_{ref} . At time $t = 60$ s the coordinate x_{ref} is switched. The time of the transition process along the coordinate x is determined by the distance to the target location and amounts to a few seconds. The standard deviation of the coordinate from the given value was 0.5 m.

6. CONCLUSIONS

This paper suggests the architecture of the system for estimating the attitude and location of a multirotor vehicle in the problem of controlling trajectory motion. A software and hardware complex for an automatic flight control system has been created. The efficiency of the suggested control system has been experimentally confirmed. The achieved positioning accuracy of the quadcopter relative to the desired trajectory when flying indoors was 0.05 m.

FUNDING

The work was supported by the Ministry of Science and Higher Education of the Russian Federation (project no. 121042900050-6).

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing UAV," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Nice, France, 2008* (IEEE, 2008), pp. 340–345. <https://doi.org/10.1109/IROS.2008.4650766>
2. J. Calusdian, X. Yun, and E. Bachmann, "Adaptive-gain complementary filter of inertial and magnetic data for orientation estimation," in *IEEE Int. Conf. on Robotics and Automation, Shanghai, 2011* (IEEE, 2011), pp. 1916–1922. <https://doi.org/10.1109/ICRA.2011.5979957>
3. S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *IEEE Int. Conf. on Rehabilitation Robotics, Zurich, 2011* (IEEE, 2011), pp. 1–7. <https://doi.org/10.1109/ICORR.2011.5975346>
4. S. O. H. Madgwick, S. Wilson, R. Turk, J. Burrige, C. Kapatos, and R. Vaidyanathan, "An extended complementary filter for full-body MARG orientation estimation," *IEEE/ASME Trans. Mechatronics* **25**, 2054–2064 (2020). <https://doi.org/10.1109/TMECH.2020.2992296>
5. A. Deibe, J. A. A. Nacimiento, J. Cardenal, and F. L. Pena, "A Kalman filter for nonlinear attitude estimation using time variable matrices and quaternions," *Sensors* **20**, 6731 (2020). <https://doi.org/10.3390/s20236731>
6. W. Li, Zh. Zheng, and S. Ping, "Quaternion-based Kalman filter for AHRS using an adaptive-step gradient descent algorithm," *Int. J. Adv. Robot. Syst.* **12**, 1–12 (2015). <https://doi.org/10.5772/61313>

7. S. P. H. Driessen, N. H. J. Janssen, L. Wang, J. L. Palmer, and H. Nijmeijer, “Experimentally validated extended Kalman filter for UAV state estimation using low-cost sensors,” *IFAC-PapersOnLine* **51** (15), 43–48 (2018). <https://doi.org/10.1016/j.ifacol.2018.09.088>
8. D. M. Bevly and B. Parkinson, “Cascaded Kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles,” *IEEE Trans. Control Syst. Technol.* **15**, 199–208 (2007). <https://doi.org/10.1109/TCST.2006.883311>
9. R. W. Beard, “Quadrotor dynamics and control,” Preprint (Brigham Young Univ., 2008).
10. K. Yu. Kotov, A. S. Mal'tsev, A. A. Nesterov, and A. P. Yan, “Algorithms and architecture of the multirotor aircraft trajectory motion control system,” *Optoelectron., Instrum. Data Process.* **56**, 228–235 (2020). <https://doi.org/10.3103/S8756699020030085>
11. A. S. Dimova, K. Yu. Kotov, and A. S. Maltsev, “Trajectory control of a quadrotor carrying a cable-suspended load,” in *24th Int. Conf. on System Theory, Control and Computing (ICSTCC 2020), Sinaia, Romania, 2020* (IEEE, 2020), pp. 501–505. <https://doi.org/10.1109/ICSTCC50638.2020.9259794>
12. Yu. N. Zolotukhin, K. Yu. Kotov, A. S. Maltsev, A. A. Nesterov, M. N. Filippov, and A. P. Yan, “Correction of transportation lag in the mobile robot control system,” *Optoelectron., Instrum. Data Process.* **47**, 141–150 (2011). <https://doi.org/10.3103/S8756699011020051>
13. Y. S. Pyo, H. C. Cho, R. W. Jung, and T. H. Lim, *ROS Robot Programming* (ROBOTIS, Seoul, 2017).

Translated by L.Trubitsyna