

УДК 681.5

АЛГОРИТМЫ И АРХИТЕКТУРА СИСТЕМЫ УПРАВЛЕНИЯ ТРАЕКТОРНЫМ ДВИЖЕНИЕМ МУЛЬТИРОТОРНОГО ЛЕТАТЕЛЬНОГО АППАРАТА

© К. Ю. Котов, А. С. Мальцев, А. А. Нестеров, А. П. Ян

*Институт автоматики и электрометрии СО РАН,
630090, г. Новосибирск, просп. Академика Коптюга, 1
E-mail: kotov@idisys.iae.nsk.su*

Для решения задачи отслеживания желаемой траектории по целевому положению предложен алгоритм управления на основе требуемых дифференциальных уравнений по отклонениям. Планирование маршрута осуществляется на базе авторского метода расчёта плоской траектории, состоящей из ориентированных отрезков прямых, сопряжённых клотоидами. Представлен программно-аппаратный комплекс системы автоматического управления полётом. Работоспособность системы управления в присутствии шумов измерений и внешних возмущений подтверждается результатами экспериментов по управлению полётом квадрокоптера в помещении.

Ключевые слова: управление траекторным движением, полёт по маршруту, мультироторный летательный аппарат, метод структурного синтеза.

DOI: 10.15372/AUT20200303

Введение. В отличие от спутниковых систем беспилотные летательные аппараты (БПЛА) позволяют получать снимки в условиях облачности и обеспечивают более высокое пространственное разрешение. Сегодня можно говорить об успешном опыте применения БПЛА для решения задач аэрофотосъёмки и создания цифровых карт местности. Аппараты мультироторной конструкции широко используются при решении этих задач благодаря управляемости, надёжности и простоте конструкции. Перспективными областями применения мультироторных летательных аппаратов (ЛА) являются доставка грузов, аэрогеофизические работы, точное земледелие [1–4].

Применение БПЛА сопровождается постановкой задач построения предписанных маршрутов движения и формирования управляющих воздействий, выдаваемых на тяговые двигатели, с учётом формы и размера обследуемых территорий, рельефа местности, внешних возмущений, а также динамики применяемого БПЛА.

При перемещении робота в заранее неизвестной или сложной среде осуществляется непрерывное планирование и построение траектории движения [5, 6]. Удобным в этом случае является задание траектории опорными точками и дальнейшее представление траектории в параметрическом виде [7]. При необходимости соблюдения требований ограниченности производных по времени координат целевого положения или гладкости вырабатываемых управляющих воздействий используются различные методы интерполяции желаемой траектории, например математический аппарат теории сплайнов [8].

В данной работе предложен алгоритм управления траекторным движением мультироторного БПЛА на основе развиваемого метода структурного синтеза. Целевое положение ЛА определяется из процедуры планирования гладких траекторий движения, заданных последовательностью поворотных пунктов маршрута с использованием кривых Корню.

Создан прототип системы управления автономным полётом. Проведённые эксперименты по управлению полётом квадрокоптера в помещении показали работоспособность предложенной системы управления.

Постановка задачи. Положение мультироторного ЛА в пространстве характеризуется координатами x, y, z центра масс аппарата в неподвижной декартовой системе и углами Эйлера ψ, φ, θ , определяющими ориентацию связанной с осями ЛА системы координат по отношению к неподвижной системе xyz [9].

Упрощённые уравнения динамики, описывающие движение в указанных координатах, имеют следующий вид:

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

Здесь m — масса ЛА, g — ускорение свободного падения, u_1 — суммарная сила тяги. Здесь и далее точками над знаками переменных обозначаются производные по времени.

По причине того, что для установления заданной ориентации аппарата используется встроенная система управления полётного контроллера, будем считать, что процессы изменения углов ориентации представимы в виде независимых дифференциальных уравнений второго порядка [10]:

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

Величины коэффициентов T и d оценены экспериментально.

Кинематическая схема ЛА определяет пересчёт силы тяги приводов в управляющие моменты по соответствующим осям вращения. Полагаем, что данный пересчёт заложен в полётном контроллере, а управляющими воздействиями являются потребная тяга u_1 и ориентация, задаваемая углами $\psi_{ref}, \varphi_{ref}, \theta_{ref}$.

Задачей управления будем считать траекторное управление движением аппарата, а именно вывод центра масс аппарата в целевое положение с координатами $x_{ref}, y_{ref}, z_{ref}$ и дальнейшее следование за ним. Необходимо найти такие управляющие воздействия $u_1, \psi_{ref}, \varphi_{ref}, \theta_{ref}$, которые обеспечат сведение к нулю рассогласования между текущими и целевыми координатами. Предполагается, что известны координаты центра масс ЛА и его ориентация в пространстве в каждый момент времени, а также первые производные этих величин. Также считается, что тяговые двигатели обладают достаточной мощностью.

Синтез алгоритма управления. Методы интерполяции предписанной траектории, например аппарат сплайнов [8] или теория спиралей Корню [11], позволяют определить производные по времени координат целевого положения на построенном маршруте. С учётом этого запишем требуемые дифференциальные уравнения на отклонение центра масс аппарата от целевого положения $x_{ref}, y_{ref}, z_{ref}$:

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

Определим вынужденное движение к желаемой траектории ($S_x = 0, S_y = 0, S_z = 0$) уравнениями

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

Здесь k_x, k_y, k_z и $\alpha_x, \alpha_y, \alpha_z$ — положительные постоянные коэффициенты, определяющие время выхода в целевое положение. Синтезируемая на основе уравнений (3), (4) система является инвариантной относительно формы задания $x_{ref}, y_{ref}, z_{ref}$ и обладает свойством астатизма.

Подставляя (3) в (4), получим

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

Учитывая (1), (2), исключим старшие производные регулируемых величин в (5):

$$\begin{cases} (\sin \psi \cdot \sin \varphi + \cos \psi \cdot \cos \varphi \cdot \sin \theta) u_1 = -m A_x; \\ (-\cos \psi \cdot \sin \varphi + \sin \psi \cdot \cos \varphi \cdot \sin \theta) u_1 = -m A_y; \\ \cos \varphi \cdot \cos \theta u_1 = -m A_z, \end{cases} \quad (6)$$

где

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

Возведём в квадрат левые и правые части уравнений (6) и, сложив их почленно, получим необходимую суммарную силу тяги двигателей:

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

Система уравнений (6) не является независимой. Это объективно отражает тот факт, что для управления курсом мультироторного ЛА достаточно управления любыми двумя углами из ψ, φ, θ . При управлении, например, углами тангажа и крена угол рыскания ψ_{ref} можно выбирать из других соображений, не связанных с изменением направления движения. Согласно методике, изложенной в [12], из уравнений (6) найдём необходимые значения углов тангажа θ_{ref} и крена φ_{ref} :

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

При положительных значениях коэффициентов α и k полученными управляющими воздействиями обеспечиваются устойчивые переходные процессы, а при $\alpha = k$ процессы будут иметь монотонный характер. Решения этих уравнений имеют одинаковый вид для различных законов изменения задающих воздействий.

Входными данными для алгоритма управления являются зашумлённые реализации координат центра масс ЛА. Дополнительно требуется наличие оценок первых производных этих переменных. Соотношения (1), (2) могут быть использованы для оценки переменных состояния объекта в расширенном фильтре Калмана. Дополним модель процесса

(1), (2) моделью наблюдений и запишем полученную нелинейную динамическую систему в компактном виде:

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

где $X = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \psi, \varphi, \theta, \dot{\psi}, \dot{\varphi}, \dot{\theta}]^\top$; $U = [u_1, \psi_{ref}, \varphi_{ref}, \theta_{ref}]^\top$; f — нелинейная вектор-функция, соответствующая правой части уравнений (1), (2); W — входной шум; V — шум измерений.

В расширенном фильтре Калмана нелинейная система (10) линеаризуется посредством разложения функций f и h в ряд Тейлора. Дальнейшая оценка вектора состояния X состоит из последовательных этапов экстраполяции и коррекции [10].

В качестве измеряемых компонент вектора X использовались координаты AR (Augmented Reality) маркера, закреплённого в центре масс аппарата, углы рыскания, крена, тангажа и угловые скорости вращения аппарата, вычисленные бортовой инерциальной системой. Результирующий вектор измерения имеет вид $Z = [x_{cam}, y_{cam}, z_{cam}, \psi_{imu}, \varphi_{imu}, \theta_{imu}, \dot{\psi}_{imu}, \dot{\varphi}_{imu}, \dot{\theta}_{imu}]^\top$. В отличие от работ [10, 13] здесь, в векторе Z , отсутствуют компоненты линейной скорости $\dot{x}_{cam}, \dot{y}_{cam}, \dot{z}_{cam}$. Применение этих величин, вычисленных посредством фильтрации сигналов $x_{cam}, y_{cam}, z_{cam}$ приводит к возникновению низкочастотных колебаний в системе управления (9), вызванных запаздыванием, возникающим при фильтрации.

Описание программной и аппаратной частей системы управления.

Разработанная система управления имеет двухуровневую структуру. Обмен данными между контроллерами верхнего и нижнего уровней реализован по протоколу mavlink через последовательный интерфейс UART (рис. 1). В качестве управляющего контроллера нижнего уровня используется автопилот Pixhawk с 32-битным STM32F427 Cortex M4 core процессором и операционной системой NuttX RTOS с открытым исходным кодом. Оценка углового положения аппарата выполняется в модуле комплементарного фильтра программного обеспечения (ПО) PX4 [14] с использованием данных интегрированного инерциального сенсора MPU 6000. Установление заданной ориентации выполняется встроенным в ПО PX4 модулем управления на основе линейных регуляторов. Переключение режимов работы автопилота производится с пульта радиоуправления (РУ). В автоматическом режиме на вход автопилота подаётся управляющая команда `set_attitude`, содержащая требуемые значения углов Эйлера и суммарной тяги двигателей. Взлёт и посадка аппарата осуществляется в ручном режиме по командам управления от пульта РУ. Абсолютные координаты центра масс аппарата при полётах в помещении определяются с помощью внешних видеокамер и системы локализации на основе визуальных AR-маркеров. Расчёт управляющих воздействий с учётом заданного маршрута полёта и текущих координат аппарата, комплексование и фильтрация данных сенсоров выполняются на верхнем уровне — одноплатном микрокомпьютере Raspberry Pi 3B с 4-ядерным процессором Cortex-A53 (ARM v8) 1,2 ГГц и оперативной памятью 1 ГБ. В качестве операционной системы используется Ubuntu 16.04 LTS.

Программное обеспечение созданной системы управления имеет модульную архитектуру и построено на механизме межпроцессного взаимодействия ROS (Robot Operation System) [15]. В данной работе используются пакеты программ ROS, предназначенные для решения задач коммуникации между контроллерами, распознавания AR-маркеров, сохранения данных. Разработанные авторами модули управления и оценки состояния в расширенном фильтре Калмана написаны на языке Python с применением асинхронной обработки на основе ROS Timer и интегрированы в систему ROS.

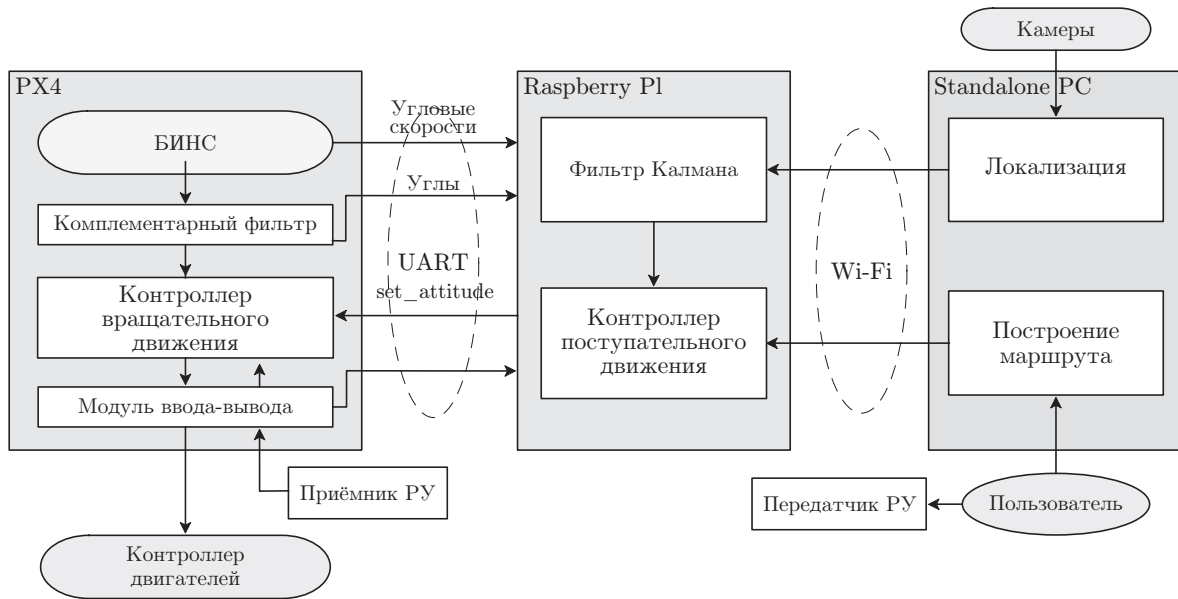


Рис. 1. Структурная схема системы управления

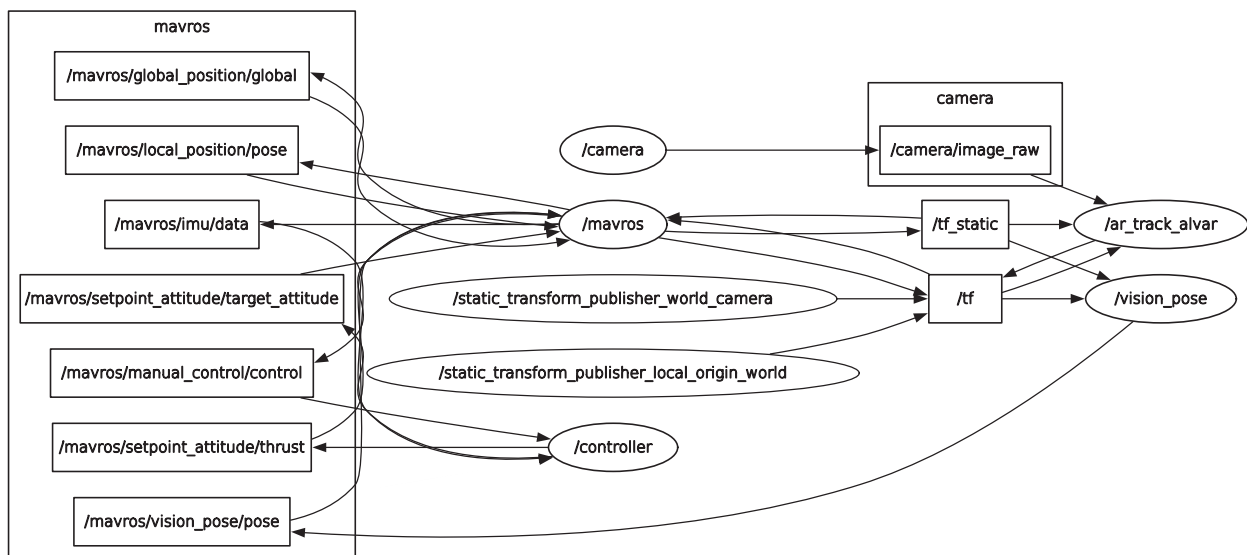


Рис. 2. Структурная схема программной части системы управления

Разработанная система управления полётом включает набор независимых программ, взаимодействующих между собой посредством обмена сообщениями в системе ROS. Её структура показана на рис. 2. В состав системы входят программные модули, выполняющие следующие функции:

- получение изображения с камеры (camera);
- распознавание AR-маркеров (ar_track_alvar);
- определение пространственного положения маркеров (vision_pose);
- управление траекторным движением и оценка вектора состояния в расширенном фильтре Калмана (controller);
- коммуникация с полётным контроллером PX4 (mavros).

Абсолютные координаты центра масс ЛА в пространстве, углы ориентации и угловые скорости вращения относительно неподвижной системы координат передаются в

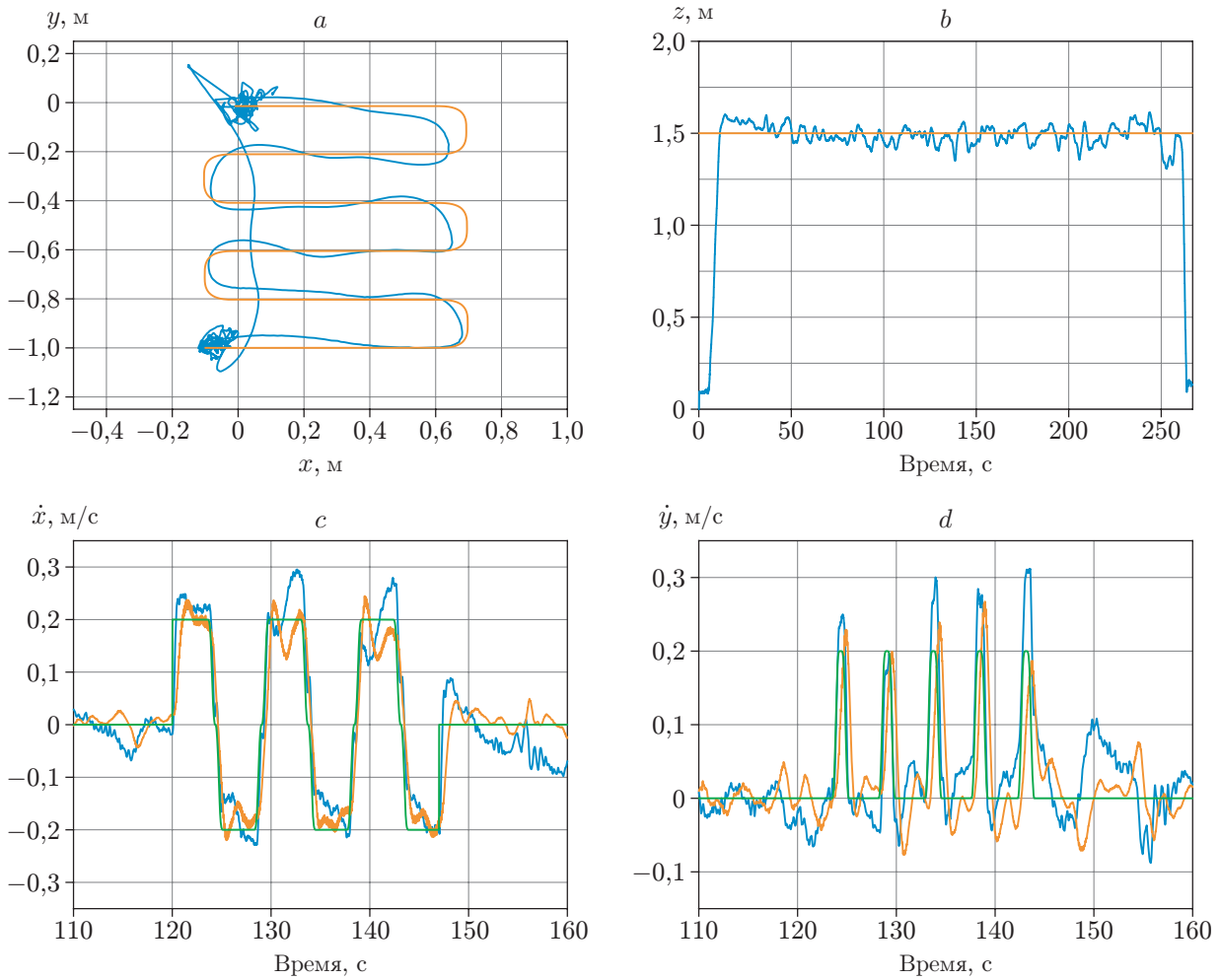


Рис. 3. Движение вдоль траектории, заданной отрезками прямых и клотоид: *a* — траектория движения квадрокоптера в плоскости (x, y) ; *b* — высота z над поверхностью; *c* — оценка линейной скорости \dot{x} ; *d* — оценка линейной скорости \dot{y}

расширенный фильтр Калмана. Полученные оценки координат используются программой controller, которая вычисляет управляющие команды для требуемых углов ориентации и суммарной тяги двигателей в соответствии с разработанным алгоритмом траекторного движения. Затем эти команды через модуль mavros поступают для исполнения в полётный контроллер PX4.

Экспериментальные результаты. Метод планирования траектории движения с применением клотоиды для перехода между отрезками прямых маршрута, заданного последовательностью поворотных пунктов, реализован в среде MATLAB/Simulink [10]. Сгенерированная программная траектория поступает на вход управляющего алгоритма (9). При построении траектории использованы следующие значения параметров: линейная скорость движения цели $V = 0,2; 0,4$ м/с; максимально допустимая нормальная перегрузка $n_{enable} = 0,08; 0,3$.

В уравнениях движения ЛА (1), (2) установлены следующие параметры: $m = 0,4$ кг, $T_\psi = T_\varphi = T_\theta = 0,17$ с, $d_\psi = d_\varphi = d_\theta = 0,35$, а для системы управления: $k_x = \alpha_x = 4,0$, $k_y = \alpha_y = 4,0$, $k_z = \alpha_z = 4,0$.

Подача управляющих команд setpoint_attitude выполнялась с шагом дискретизации 0,01 с. Данные об ориентации аппарата, вычисленные его навигационной системой, а так-

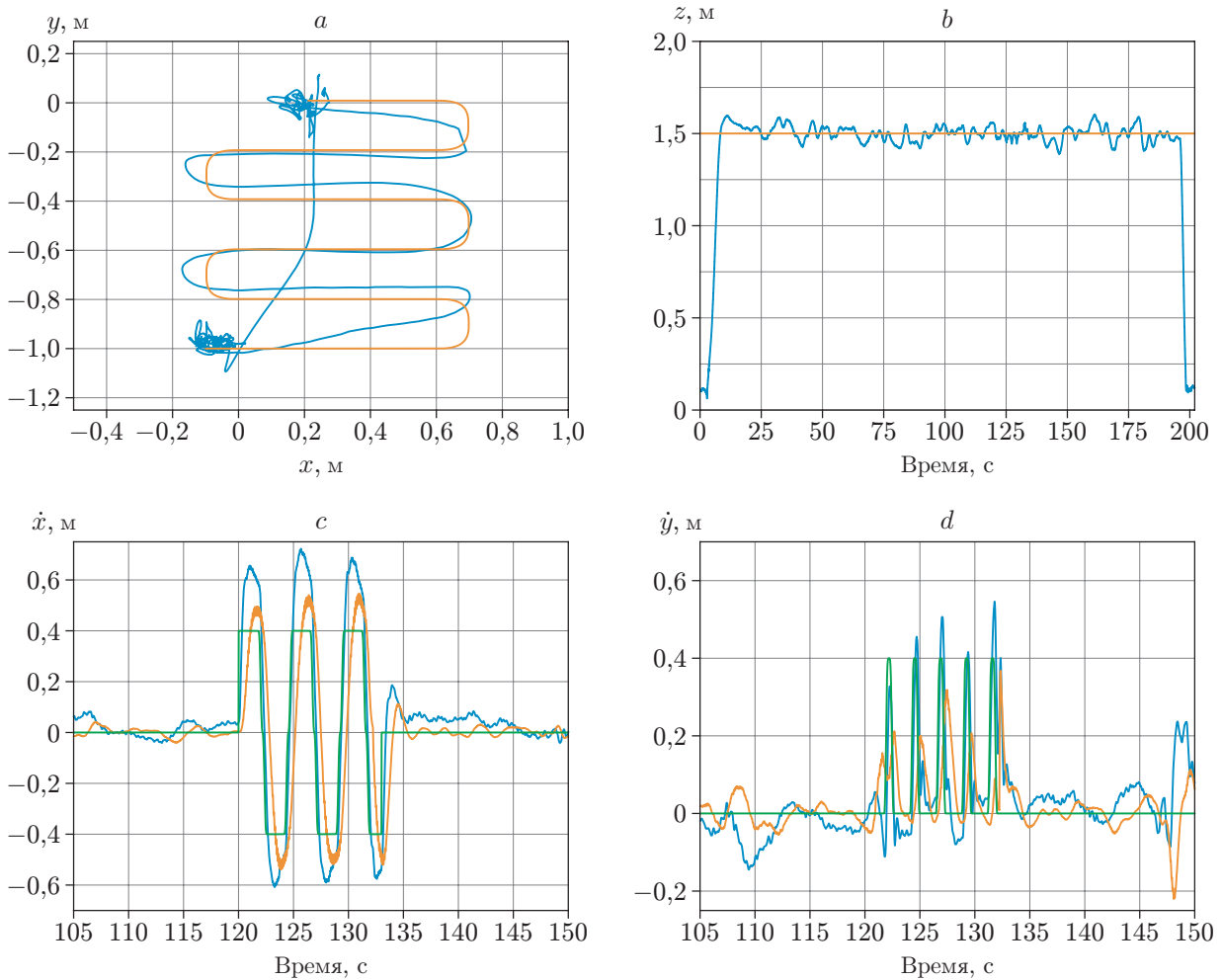


Рис. 4. Движение вдоль траектории, заданной отрезками прямых и клотоид (обозначения соответствуют рис. 3)

же координаты центра масс аппарата от внешней видеосистемы поступали с интервалами 0,02 и 0,05 с соответственно.

Перед началом движения аппарат обрабатывал команду на удержание неподвижного целевого положения до момента времени $t = 120$ с. Момент времени $t = 0$ с соответствует моменту запуска управляющей программы controller. Рассчитанное значение среднеквадратичного отклонения от средних переменных x, y, z составило 0,03 м. Расчёт выполнен на интервале времени от 30 до 115 с с отбрасыванием значений, превышающих по модулю 3,5 от первоначально вычисленного.

На рис. 3 приведены результаты для случая движения вдоль траектории с параметрами $V = 0,2$ м/с, $n_{enable} = 0,08$. Отклонение от траектории во время движения составило 0,05 м.

Оценка компонент поступательной скорости \dot{x}, \dot{y} , полученная в фильтре Калмана, сравнивалась с оценкой линейных скоростей, полученных посредством фильтрации сигналов x, y . Разница в оценке составила 0,1 м/с (рис. 3, c, d).

Увеличение скорости перемещения вдоль программной траектории до $V = 0,4$ м/с и перегрузки до $n_{enable} = 0,3$ приводит к снижению точности выдерживания заданного положения при манёврах (рис. 4).

Заключение. В данной работе предложен алгоритм траекторного управления движением мультироторного аппарата в задаче отслеживания целевого положения. Создан программно-аппаратный комплекс системы автоматического управления полётом. Экспериментально подтверждена эффективность предложенного ранее метода планирования маршрута с использованием кривых Корню, обеспечивающего плавное изменение перегрузки на поворотах. Достигнутая точность позиционирования квадрокоптера относительно желаемой траектории при полётах в помещении составила 0,05–0,1 м.

Финансирование. Работа выполнена при поддержке Министерства науки и высшего образования РФ (государственная регистрация № АААА-А17-117060610006-6).

СПИСОК ЛИТЕРАТУРЫ

1. **Kuckelhaus Dr. M., Niezgoda D.** Unmanned Aerial Vehicles in Logistics: Tech. rep. DHL, 2014. 24 p.
2. **Duggal V., Sukhwani M., Bipin K. et al.** Plantation monitoring and yield estimation using autonomous quadcopter for precision agriculture // Proc. of the IEEE Intern. Conference on Robotics and Automation (ICRA). Stockholm, Sweden, May 16–21, 2016. P. 5121–5127.
3. **Cwiakala P., Kocierz R., Puniach E. et al.** Assessment of the possibility of using unmanned aerial vehicles (uavs) for the documentation of hiking trails in alpine areas // Sensors. 2017. **18**, N 1. P. 81.
4. **Zhang Ch., Kovacs J. M.** The application of small unmanned aerial systems for precision agriculture: a review // Precision Agriculture. 2012. **13**, N 6. P. 693–712.
5. **Brock O., Khatib O.** Real-time replanning in high-dimensional configuration spaces using sets of homotopic paths // Proc of the Intern. Conference on Robotics and Automation (ICRA). IEEE, 2000. P. 550–555.
6. **Першина Ж. С., Каздорф С. Я., Лопота А. В.** Методы визуальной навигации мобильного робота и построения картографических моделей внешней среды // Автометрия. 2019. **55**, № 2. С. 92–102. DOI: 10.15372/AUT20190210.
7. **Thrun S., Montemerlo M., Dahlkamp H. et al.** Stanley: The robot that won the darpa grand challenge // Journ. Field Robotics. 2006. **23**, N 9. P. 661–692.
8. **Connors J., Elkaim G.** Analysis of a spline based, obstacle avoiding path planning algorithm // VTC Spring. IEEE, 2007. P. 2565–2569.
9. **Kim J., Kang M.-S., Park S.** Accurate modeling and robust hovering control for a quad—rotor vtol aircraft // Journ. Intell. Robotics Syst. 2010. **57**, N 1–4. P. 9–26.
10. **Белоконь С. А., Золотухин Ю. Н., Котов К. Ю. и др.** Использование фильтра Калмана в системе управления траекторным движением квадрокоптера // Автометрия. 2013. № 6. С. 14–24.
11. **Белоконь С. А., Золотухин Ю. Н., Нестеров А. А.** Планирование маршрутов движения летательного аппарата с использованием гладких траекторий // Автометрия. 2017. **53**, № 1. С. 3–11. DOI: 10.15372/AUT20170101.
12. **Белоконь С. А., Золотухин Ю. Н., Мальцев А. С.** Управление параметрами полёта квадрокоптера при движении по заданной траектории // Автометрия. 2012. № 5. С. 32–41.
13. **Димова А. С., Котов К. Ю., Мальцев А. С. и др.** Управление квадрокоптером в задаче транспортировки груза на подвесе // Автометрия. 2018. **54**, № 5. С. 116–121. DOI: 10.15372/AUT20180515.
14. **Meier L., Honegger D., Pollefeys M.** Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms // Proc. of the IEEE Intern. Conference on Robotics and Automation (ICRA). Seattle, USA, May 26–30, 2015. P. 6235–6240.

15. **Quigley M., Conley K., Gerkey B. P. et al.** Ros: an open-source robot operating system // ICRA Workshop on Open Source Software. 2009. Vol. 3. P. 5.

Поступила в редакцию 30.12.2019

После доработки 05.02.2020

Принята к публикации 17.02.2020
