

Algorithm guidance for movement with possibility of quadcopters avoiding obstacles and tracking planned route based on the normal acceleration controlling

Geng K.¹, Chulin N.²

Алгоритм наведения движения для квадрокоптера с возможностью облета препятствий и отслеживания запланированного маршрута на основе управления нормальным ускорением

Гэн К.¹, Чулин Н. А.²

¹Гэн Кэкэ / Geng Keke – аспирант;

²Чулин Николай Александрович / Chulin Nikolai – кандидат педагогических наук, доцент, кафедра систем автоматического управления, факультет информатики и систем управления, Московский государственный технический университет имени Н. Э. Баумана, г. Москва

Аннотация: в работе предлагаются алгоритмы наведения для квадрокоптера с возможностью облета препятствий и отслеживания запланированного маршрута на основе управления нормальным ускорением. Построена математическая модель квадрокоптера. Представленная разработка многорежимного контроллера содержит набор бэкстепинг-регуляторов, автоматически выбираемых в соответствии с условиями полёта. Разработаны улучшения муравьиного алгоритма, позволяющие повысить скорость вычислений и найти маршрут с меньшим количеством точек на глобальном маршруте. Упрощены объёмы вычисления алгоритма отслеживания маршрута при выборе точки маршрута в качестве опорной точки. Представлена разработка улучшения алгоритма отслеживания маршрута с адаптивным выбором точки маршрута при создании функции выбора опорной точки. В соответствии с экстремумом функции расстояния между роботом и движущимися препятствиями текущего момента предсказана позиция столкновения, и создано виртуальное статическое препятствие в этой позиции в качестве мишени облета. Вычислено минимальное боковое ускорение избегания столкновения, для повышения безопасности необходимо умножить на коэффициент облета. Результаты моделирования показывают правильность и обоснованность предлагаемого метода.

Abstract: we proposed guidance algorithms for quadcopters with possible detour obstacles and tracking the planned route based on the normal acceleration control. A mathematical model of quadcopters was built. Present the development of multi-mode controller provides a set of backstepping controllers are automatically selected according to the flight conditions. An improved ant algorithm was proposed, allowing to increase computing speed and find the route with less waypoints on the global route. Simplified volume calculation algorithm of tracing the route with adaptive choosing a route point as a reference point. In accordance with the extremum function of the distance between the robot and moving obstacles of the moment, predicted collision position, and created a virtual static obstacle in this position as a target flyby. Calculate the minimum lateral acceleration collision avoidance, to increase safety must be multiplied by the coefficient of detour. The simulation results show the correctness and validity of the proposed method.

Ключевые слова: квадрокоптер, отслеживание маршрута, планирование маршрута, муравьиный алгоритм, облета препятствий, фильтр Калмана.

Keywords: quadcopter, route tracking, route planning, ant algorithm, avoiding obstacles, Kalman filter.

УДК 004.021

Введение

Квадрокоптер представляет собой беспилотный летательный аппарат (БПЛА), имеющий четыре двигателя с воздушными винтами (пропеллерами), создающими тягу. В настоящее время подобные аппараты используются достаточно широко и разнообразно, но это использование ограничено, в основном, режимами «ручного» дистанционного управления с пульта оператора.

Актуальной является задача разработки системы управления, позволяющая осуществлять автономный полёт квадрокоптера по заданному маршруту. В публикациях по данному вопросу задача рассматривается как сочетание задач управления угловым положением, траекторным движением, облетом препятствий и отслеживанием запланированного маршрута. Обсуждаются и предлагаются различные подходы, в частности использование линейных квадратичных регуляторов, успешно решающих задачи стабилизации при зависимости, но не работающих при существенных нелинейностях и перекрёстных связях [1], использование скользящего режима управления, простого и надёжного, но требующего адаптации логики переключений к режимам полёта [2]. В работе [3] предложен алгоритм отслеживания траектории квадрокоптера на основе ПД регулятора и метода «бэкстепинг» с корректирующим фильтром. В работах [4, 5] предлагается нелинейный контроллер на основе нейронной сети, а в [6] - алгоритм «реального времени» планирования траектории и предотвращения столкновений. В работе [7] рассматривается

адаптивный алгоритм отслеживания положения для вертикального взлета и посадки при ограниченных внешних возмущениях. Планирование глобального исключаящего столкновения маршрута является одним из основных вопросов при работе движущихся роботов, как наземных, так и воздушных. Есть много методов, предлагаемых для решения задачи планирования маршрута. Для большинства известных методов, таких как клеточное разбиение [8], дорожная карта [9], потенциальное поле [10], затруднено решение проблем планирования в сложных условиях и адаптации, не решен вопрос устойчивости. Муравьиный алгоритм, обладающий параллельностью и положительной обратной связью, в последние годы широко используется для планирования маршрута в статической и динамичной среде [11]. Отслеживание траектории является очень важной частью в системе управления автономным полетом. Отслеживание траектории в основном является процессом стремления поперечного расстояния и касательной углы к нулю и непрерывное движение по заданной траектории [12]. Два этапа можно рассматривать для решения задачи отслеживания заданного маршрута. Первый этап является генерированием ускорения движения [13]. Второй этап является управлением полетом квадрокоптера, чтобы следовать командам генерированного ускорения системе управления траекторным движением [14].

В статической среде есть много методов, предлагаемых для решения задачи планирования маршрута. Расширение проблемы планирования маршрута в динамической среде тоже было тщательно исследовано. Нечеткая логика достаточно широко используется для обхода подвижных препятствий при управлении направлением движения робота и скоростью [15]. Чтобы улучшить прочность нечеткой логики при планировании маршрута в динамической среде, часто используются вместе с другими алгоритмами [16-17]. Нейронная сеть тоже является осуществимой для избегания столкновения с подвижными препятствиями при непрерывности проб и изучения локальной окружающей среды [18-19]. Китайские исследователи представили метод с использованием подвижного окна для избегания столкновений в режиме реального времени в неизвестной среде [20].

В этой статье предлагается метод наведения движения для квадрокоптера с возможностью облета препятствий и отслеживания запланированного маршрута. Блок-схема метода показана на рис. 1. На рисунке видно, что наш метод состоит в следующем: блок I - глобального планирования маршрута; блок II - облета препятствий; блок III - отслеживания запланированного маршрута; блок IV - угловые и траекторные управления движением квадрокоптера.

Блок стабилизации полета и траекторного управления полетом квадрокоптера. Для более точного описания состояния полета квадрокоптера под управлением нашего метода, в данной работе построили математическую модель с учётом влияния экранного эффекта поверхности земли, действия ветра, гироскопических моментов двигателей и винтов.

Аэродинамический анализ винта подветрового возмущения показан на рис. 1.

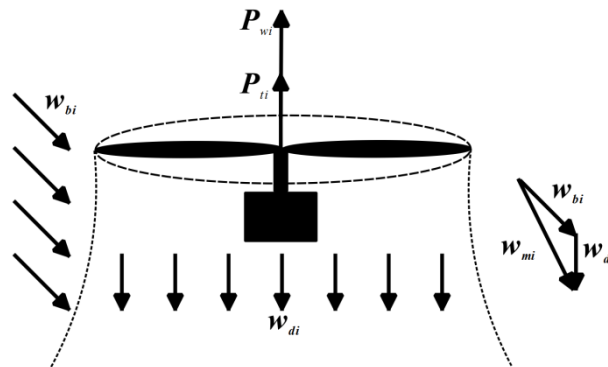


Рис. 1. Аэродинамический анализ винта подветрового возмущения

На рисунке для i -го винта: w_{bi} - вектор ветра в связанной системе координат, w_{di} - индуцированная скорость, w_{mi} - результирующая индуцированная скорость винта, P_{ii} - сила тяги винта, P_{wi} - сила тяги ветра.

Сила тяги и индуцированная скорость винта можно записать следующим образом [21-22]:

$$P_{ii} = c_{pi} \frac{\omega_i^2}{2} \rho S_{pi}; w_{di} = \sqrt{\frac{P_{ii}}{2\rho S_i}} \quad (1)$$

где c_{pi} - коэффициент силы тяги, ρ - плотность воздуха, S_i - площадь диска винта, S_{pi} - проекция площади винта.

Результирующая индуцированная скорость винта:

$$w_{mi} = w_{di} + w_{bi} \quad (2)$$

По литературе [22] сила тяги i -го винта подветрового возмущения:

$$P_{w_i} = P_{ii} + P_{wi} = 2\rho S_i w_{di} w_{mi} \quad (3)$$

При полете вблизи поверхности земли эффект земли является очевидным, то есть на диск винта будет действовать дополнительная сила тяги, что означает, что самолет требует меньше энергии для поддержания состояния полета.

Отношение силы тяги без учета эффекта земли и силы тяги с учетом эффекта земли [23]:

$$\frac{P_{gi}}{P_{ii}} = \frac{1}{1 - \left(\frac{r_i}{4h}\right)^2 \left[1 / \left(1 + \left(\frac{V}{w_{di}}\right)^2\right)\right]} \quad (4)$$

где h - высота полета; r_i - радиус диска i -го винта; V - скорость полета в нормальной земной системе координат.

Для i -го винта результирующая сила тяги:

$$P_w = \left[0, \sum_1^4 P_i, 0\right]^T; P_i = P_{ii} + P_{wi} + P_{gi} \quad (5)$$

Сила сопротивления воздуха может быть получена с использованием уравнения динамического давления [24]:

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \frac{1}{2} \rho \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} \begin{bmatrix} V_{wx}^2 & 0 & 0 \\ 0 & V_{wy}^2 & 0 \\ 0 & 0 & V_{wz}^2 \end{bmatrix} \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} \quad (6)$$

где $[S_x, S_y, S_z]^T$ - площадь квадрокоптера подвергающаяся воздействию ветра; $[V_{wx}, V_{wy}, V_{wz}]^T$ - скорость ветра; $[C_x, C_y, C_z]^T$ - коэффициент сопротивления.

Площадь квадрокоптера, подвергающуюся воздействию ветра, можно записать следующим образом:

$$\begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} 4P_s \pi^2 \sin \eta + S_0 \\ 4P_s \pi^2 \cos \eta + S_0 \\ 4P_s \pi^2 \sin \eta + S_0 \end{bmatrix} \quad (7)$$

где P_s - коэффициент, который выражает полупроницаемых характер вращающимся винтом; η - угол наклона (сочетание тангажа и крена углов); S_0 - площадь корпуса квадрокоптера подвергающаяся воздействию ветра.

Сила сопротивления воздуха и сила тяжести:

$$P_f = [-f_x, -f_y, -f_z]^T; G = [0, mg, 0]^T \quad (8)$$

где m - масса квадрокоптера; g - ускорение силы тяжести.

Сила тяги в нормальной земной системе координат:

$$P = [P_x, P_y, P_z]^T = R P_w + P_g + P_f \quad (9)$$

где R - матрица перехода; ψ, ϑ, γ - углы рыскания, тангажа, крена.

$$R(\psi, \vartheta, \gamma) = \begin{bmatrix} \cos \psi \cos \vartheta & -\cos \psi \sin \vartheta \cos \gamma + \sin \psi \sin \gamma & \cos \psi \sin \vartheta \sin \gamma + \sin \psi \cos \gamma \\ \sin \vartheta & \cos \vartheta \cos \gamma & \cos \vartheta \sin \gamma \\ -\sin \psi \cos \vartheta & \sin \psi \sin \vartheta \cos \gamma + \cos \psi \sin \gamma & -\sin \psi \sin \vartheta \sin \gamma + \cos \psi \cos \gamma \end{bmatrix} \quad (10)$$

Уравнения динамики движения центра масс в нормальной земной системе координат:

$$\ddot{x} = P_x / m; \ddot{y} = P_y / m; \ddot{z} = P_z / m \quad (11)$$

С учетом симметрии аппарата и считая, что центр масс расположен в начале координат связанной системы, уравнения динамики углового движения в связанной системе координат можно записать в виде:

$$\begin{cases} \dot{w}_x = w_y w_z (I_y - I_z) / I_x + M_{Rx} / I_x \\ \dot{w}_y = w_x w_z (I_z - I_x) / I_y + M_{Ry} / I_y \\ \dot{w}_z = w_x w_y (I_x - I_y) / I_z + M_{Rz} / I_z \end{cases}; \begin{cases} M_{Rx} = M_{qx} + M_{mx} + M_{px} + M_{wx} + M_{gx} \\ M_{Ry} = M_{qy} + M_{wy} + M_{gy} \\ M_{Rz} = M_{qz} + M_{mz} + M_{pz} + M_{wz} + M_{gz} \end{cases} \quad (12)$$

где w_x, w_y, w_z - проекции вектора угловой скорости аппарата на связанную систему координат;

M_{Rx}, M_{Ry}, M_{Rz} - проекции результирующего момента; I_x, I_y, I_z - осевые моменты инерции аппарата; M_{qx}, M_{qy}, M_{qz} - моменты, создаваемые винтами, M_{mx}, M_{mz} и M_{px}, M_{pz} - гироскопические моменты двигателей и винтов, M_{wx}, M_{wy}, M_{wz} - моменты, создаваемые ветрами, M_{gx}, M_{gy}, M_{gz} - моменты, создаваемые силами тяги эффекта земли. Если пренебречь инерционностью винтов при изменении угловых скоростей их вращения, то указанные моменты можно выразить следующим образом:

$$\begin{cases} M_{px} + M_{wx} + M_{gx} = (P_3 - P_1) \cdot l \\ M_{qy} + M_{wy} + M_{gy} = M_1 + M_2 + M_3 + M_4 \\ M_{pz} + M_{wz} + M_{gz} = (P_2 - P_4) \cdot l \\ M_i = m_{pi} w_i^2 \end{cases}; \begin{cases} M_{mx} = I_m w_z (w_1 + w_3 - w_2 - w_4) \\ M_{mz} = I_m w_x (w_2 + w_4 - w_1 - w_3) \end{cases}; \begin{cases} M_{px} = I_p w_z (w_1 + w_3 - w_2 - w_4) \\ M_{pz} = I_p w_x (w_2 + w_4 - w_1 - w_3) \end{cases} \quad (13)$$

где l - расстояние от центра масс до оси винта, I_m и I_p - моменты инерции ротора и винта; m_{pi} - коэффициент момента.

Изменения углов Эйлера определяются через проекции угловой скорости кинематическими уравнениями Эйлера:

$$\begin{cases} \dot{\psi} = (\omega_y \cos \gamma - \omega_z \sin \gamma) / \cos \gamma \\ \dot{\gamma} = \omega_x - tg \vartheta (\omega_y \cos \gamma - \omega_z \sin \gamma) \\ \dot{\vartheta} = \omega_y \sin \gamma + \omega_z \cos \gamma \end{cases} \quad (14)$$

Алгоритмы работы подсистем стабилизации и траекторного управления предлагается рассчитывать одним из известных методов, причём предпочтительнее выбирать те, которые при сравнимом качестве являются наиболее простыми. Поэтому для каждой из подсистем произведен расчёт корректирующих устройств в виде ПИД-регуляторов и методом, известным в литературе под названием «бэкстеппинг» (англ. backstepping) [25], после чего путём сравнения результатов выбран наиболее подходящий регулятор для каждой из подсистем.

Проекция горизонтальной силы имеют вид:

$$\begin{cases} U_x = P(-\cos \gamma \cos \psi \sin \vartheta + \sin \gamma \sin \psi) - f_x \\ U_z = P(\cos \gamma \sin \psi \sin \vartheta + \sin \gamma \cos \psi) - f_z \end{cases} \quad (15)$$

откуда можно определить углы крена и тангажа, при которых создаются требуемые воздействия при известной суммарной тяге P :

$$\begin{cases} \gamma_d = \arcsin \{ [(U_{zd} + f_z) \cos \psi_d + (U_{xd} + f_x) \sin \psi_d] / P \} \\ \vartheta_d = \arccos \{ [(U_{zd} + f_z) \sin \psi_d - (U_{xd} + f_x) \cos \psi_d] / P \} \end{cases} \quad (16)$$

Управляющие воздействия U_{xd} и U_{zd} , а также $U_{yd} = U_1$ для канала управления высотой, можно получить, рассматривая подсистему траекторного управления как систему регулирования, отслеживающую требуемые координаты центра масс, в частности – как выходные сигналы ПИД-регулятора по отклонениям координат центра масс от требуемых:

$$\begin{cases} U_{xd} = K_{px}(x_d - x) + K_{ix} \int (x_d - x) dt + K_{dx}(\dot{x}_d - \dot{x}) \\ U_{yd} = K_{py}(y_d - y) / \cos \gamma \cos \vartheta + K_{iy} \int (y_d - y) dt + K_{dy}(\dot{y}_d - \dot{y}) + f_z + mg \\ U_{zd} = K_{pz}(z_d - z) + K_{iz} \int (z_d - z) dt + K_{dz}(\dot{z}_d - \dot{z}) \end{cases} \quad (17)$$

Суть метода «бэкстеппинг» состоит в представлении сложной системы в виде цепочки вложенных подсистем, для каждой из которых формируются вспомогательные управляющие сигналы и составляются зависящие от этих сигналов функции Ляпунова. Выполнение критериев устойчивости по Ляпунову при последовательном выборе этих сигналов для каждой подсистемы обеспечивает устойчивость системы в целом. Процедура имеет характер пошагового обхода интеграторов обратными связями, откуда – название «integrator backstepping», или кратко - бэкстеппинг (англ. backstepping). В определённых частных случаях процедура становится регулярной и достаточно простой. Для углового движения летательного аппарата такой случай возможен при малых углах тангажа и крена, когда производные углов можно считать равными соответствующим угловым скоростям. Тогда уравнения углового движения можно приближённо представить в виде трёх подсистем

$$S_1 = \begin{cases} \dot{\gamma} = w_x \\ \dot{w}_x = w_y w_z (I_y - I_z) / I_x + M_{Rx} / I_x \end{cases}; S_2 = \begin{cases} \dot{\psi} = w_y \\ \dot{w}_y = w_x w_z (I_z - I_x) / I_y + M_{Ry} / I_y \end{cases}; S_3 = \begin{cases} \dot{\vartheta} = w_z \\ \dot{w}_z = w_x w_y (I_x - I_y) / I_z + M_{Rz} / I_z \end{cases} \quad (18)$$

Следуя приведённому алгоритму, введём для подсистемы S_1 вспомогательный управляющий сигнал

$z_1 = \gamma_d - \gamma$ и соответствующую функцию Ляпунова $V_1(z_1) = z_1^2/2$, производная которой $\dot{V}_1(z_1) = z_1 \dot{z}_1 = z_1(\dot{\gamma}_d - \dot{w}_x)$.

Второй вспомогательный управляющий сигнал сформируем в виде $z_2 = w_x - \dot{\gamma}_d - k_1 z_1$ с соответствующей функцией Ляпунова $V_2(z_1, z_2) = (z_1^2 + z_2^2)/2$, производная которой $\dot{V}_2(z_1, z_2) = z_1(\dot{\gamma}_d - \dot{w}_x) + z_2(\dot{w}_x - \ddot{\gamma}_d - k_1 \dot{z}_1)$.

Принимая для системы стабилизации $\dot{\gamma}_d = \ddot{\gamma}_d = 0$, получаем $\dot{V}_2(z_1, z_2) = z_2 \dot{w}_x + (k_1^2 - 1)z_1 z_2 - k_1 z_1^2 + k_2 z_2^2$.

Чтобы подсистема была устойчива, то есть, чтобы $\dot{V}_2(z_1, z_2) \leq 0$, причём $\dot{V}_2(z_1, z_2) = 0$ только когда $z_1 = z_2 = 0$, примем

$$\dot{V}_2(z_1, z_2) = -k_1 z_1^2 - k_2 z_2^2, k_1 > 0, k_2 > 0 \quad (19)$$

Тогда управляющий сигнал будет иметь вид:

$$U_2 = -I_x [w_y w_z (I_y - I_x) / I_x + (k_1 k_2 - 1) \gamma_d + (k_1 k_2 + 1) \gamma + (k_1 + k_2) w_x] \quad (20)$$

Функция Ляпунова $V_3(z_1) = V_2(z_1, z_2) = [z_1^2 + (w_x - \dot{\gamma}_d - k_1 z_1)^2] / 2 \geq 0$, а её производная $\dot{V}_3(z_1) = \dot{V}_2(z_1, z_2) = -k_1 z_1^2 - k_2 z_2^2 = -k_1 z_1^2 - k_2 (w_x - \dot{\gamma}_d - k_1 z_1)^2 \leq 0$, то есть замкнутая подсистема — устойчива.

Аналогичным образом можно получить другие управляющие сигналы:

$$\begin{cases} U_3 = -I_y [w_x w_z (I_x - I_y) / I_y + (k_3 k_4 - 1) \psi_d + (k_3 k_4 + 1) \psi + (k_3 + k_4) w_y] \\ U_4 = -I_z [w_x w_y (I_y - I_x) / I_z + (k_5 k_6 - 1) \vartheta_d + (k_5 k_6 + 1) \vartheta + (k_5 + k_6) w_z] \end{cases} \quad (21)$$

где $k_3 > 0$; $k_4 > 0$; $k_5 > 0$; $k_6 > 0$.

Для повышения стабильности полета, предлагается многорежимный контроллер, содержащий набор регуляторов управления угловым положением, автоматически выбираемых в соответствии с условиями полёта (высота, угловое положение, ветровые воздействия, режимы взлёта-посадки), а также – подсистему аварийного возврата к исходной точке.

Основными возмущающими факторами при полёте квадрокоптера являются ветровые воздействия и экранные эффекты при движении вблизи земли. Такие воздействия приводят к росту углов наклона аппарата в стабилизируемых режимах. Поэтому в первом приближении можно рассмотреть четыре возможных режима полёта, отличающихся настройками регуляторов. В режиме взлета или посадки лучше выбирать регулятор с меньшим перерегулированием, чтобы избежать столкновения с землей и уменьшить «дрожание» квадрокоптера из-за турбулентности экранных эффектов у поверхности земли. В полете без ветра, при слабом (3,4 - 5,4 м/с) или умеренном (5,5 - 7,9 м/с) ветре можно выбирать регуляторы с малым временем регулирования, чтобы сочетать устойчивость и маневренность. Когда ветер превышает допустимый, включается алгоритм нештатного завершения полёта, например – возврата к исходной точке в максимально «щадящем» режиме. К последнему случаю можно отнести и другие потенциально опасные ситуации (отказы оборудования, некорректное задание маршрута и т.п.).

Множество режимов полета $Q = \{q_1, q_2, q_3, q_4\}$, где q_1 : $z < z_T$, $\gamma < \gamma_{\max}$, и $\mathcal{G} < \mathcal{G}_{\max}$ - взлет и посадка; q_2 : $z \geq z_T$, $v_w < v_T$, $\gamma < \gamma_{\max}$, и $\mathcal{G} < \mathcal{G}_{\max}$ - отсутствие ветра или слабый ветер; q_3 : $z \geq z_T$, $v_T \leq v_w \leq v_{\max}$, $\gamma < \gamma_{\max}$, и $\mathcal{G} < \mathcal{G}_{\max}$ - умеренный ветер; q_4 : $v_w \geq v_{\max}$ или $\gamma \geq \gamma_{\max}$ или $\mathcal{G} \geq \mathcal{G}_{\max}$ - режим нештатного завершения полёта.

Множество соответствующих переходов $\Sigma = \{w_{12}, w_{13}, w_{14}, w_{21}, w_{23}, w_{24}, w_{31}, w_{32}, w_{34}\}$.

Диаграмма переключения режимов показана на рисунке 2.

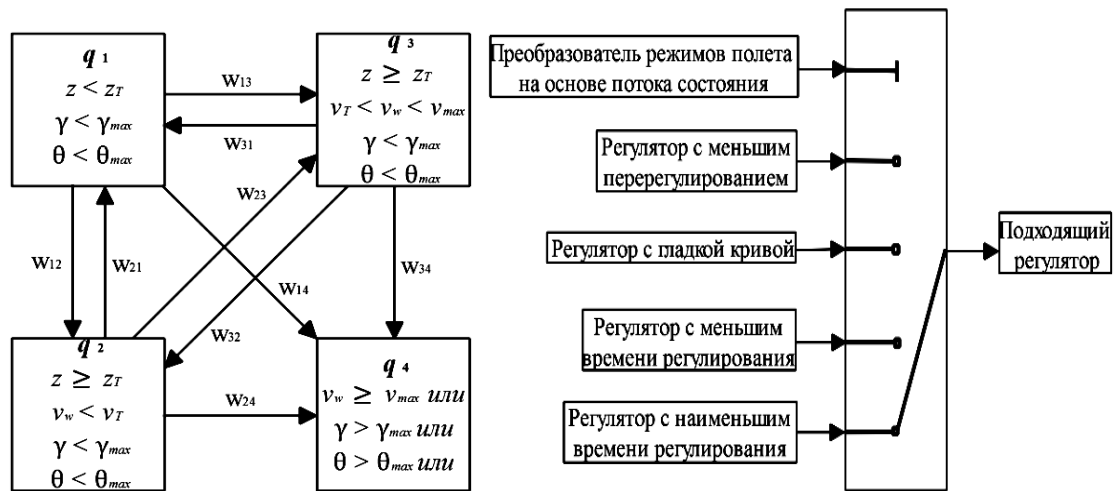


Рис. 2. Диаграмма переключения режимов

Работоспособность алгоритма проверялась моделированием реакций на ступенчатые воздействия и полёта по заданным траекториям в поле ветра, составляющем стабильный ветер, турбулентность и порывистый ветер, как показано на рис. 3.

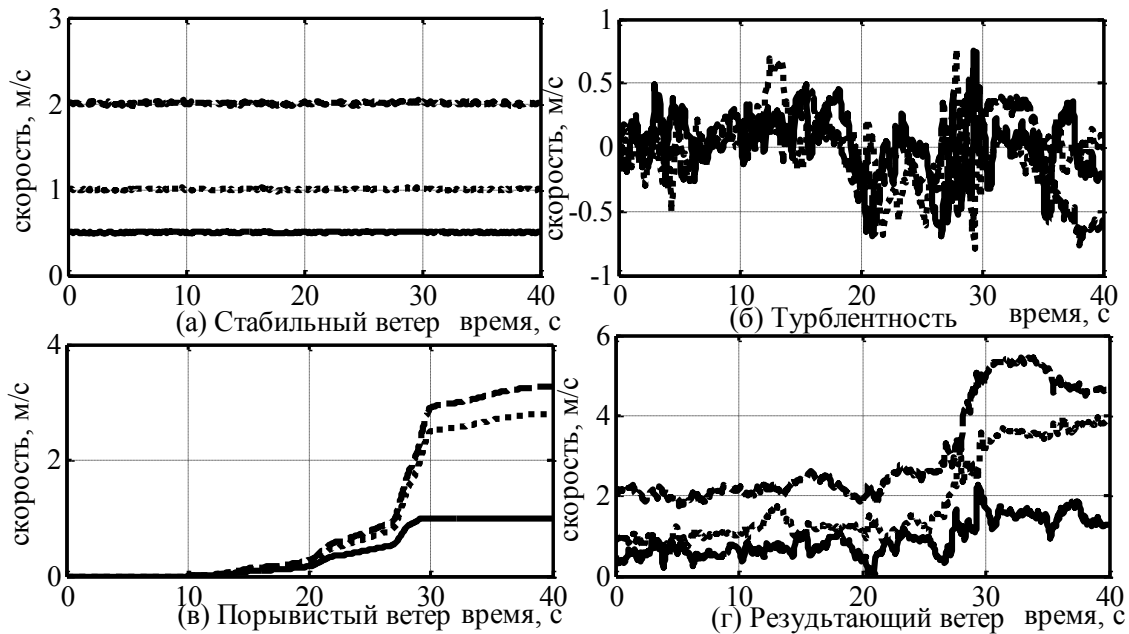


Рис. 3. Скорость ветра

На рис. 4 показаны результаты моделирования при отслеживании траектории состоящей из участков: АВ - взлет; ВС - равномерное прямолинейное движение; CD - равномерное ускоренное прямолинейное движение; DE - равномерное движение точка O_1 ; EF - равномерное прямолинейное движение; FG - равномерное движение вокруг точки O_2 ; GH - посадка.

Результаты моделирования показывают, что квадрокоптер может летать по заданному маршруту в среде без ветра и с ветром.

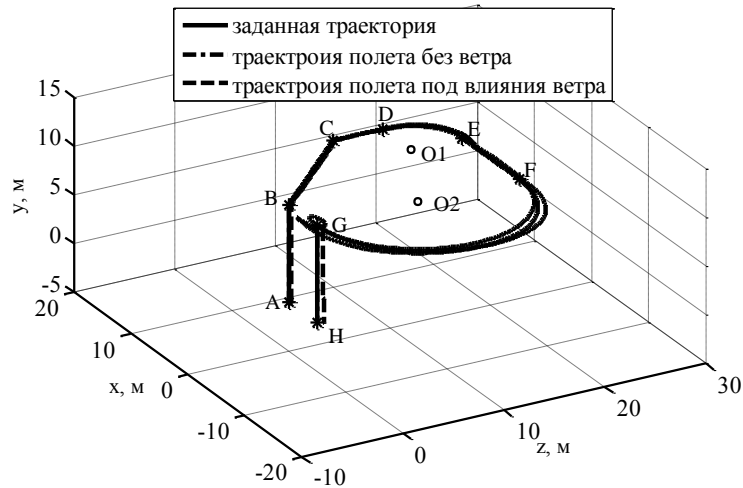


Рис. 4. Траектория полета квадрокоптера

На рис. 5 показано сравнение изменения угла наклона γ под управлением БЭК1 и многорежимного контроллера в среде без ветра и с ветром. Результаты моделирования показывают, что квадрокоптер имеет выше стабильность под управлением многорежимного контроллера.

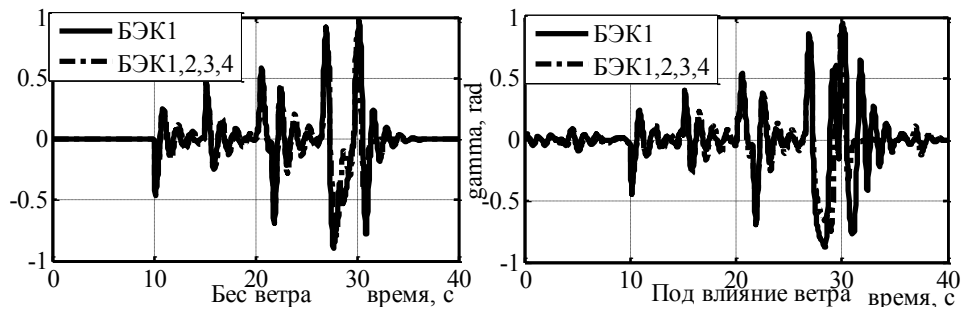


Рис. 5. Изменения угла наклона γ

Когда скорость ветра более скорости умеренного ветра, возможность аварии будет значительно увеличена, следовательно, квадрокоптер должен иметь функцию автоматического возврата на базу в суровой среде.

Блок планирования глобального маршрута. Для решения этих проблем в настоящей работе предлагается новый метод планирования маршрута на основе дискретной облачно-точечной карты.

Первым шагом планирования пути является создание поисковой карты. Карта поиска разбивается на несколько кольцевых безопасных и опасных зон в соответствии с положением и размерами препятствий с равномерным расположением точек сетки в опасных зонах, как показано на рис. 6. На этом рисунке: B_1, B_2, B_3 – препятствия; S_1, S_2, S_3 – безопасные зоны; D_1, D_2 – опасные зоны; O – позиция робота; G – целевая точка; $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow G$ – один из возможных маршрутов без столкновений.

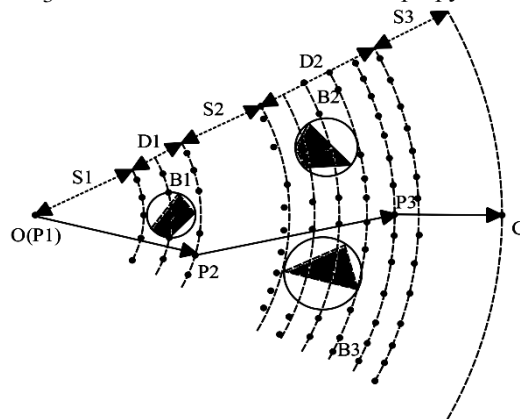


Рис. 6. Облачно-точечная карта

Видно, что структура облачно-точечной карты определяется следующим образом: $Карта = [O, G, B, S, D, P]$, где B – препятствия; S – безопасные зоны; D – опасные зоны; P – точки поиска возможных маршрутов.

Для препятствий произвольной формы, среди которых могут оказаться невыпуклые фигуры, возможны локальные минимумы, соответствующие тупиковым путям. Если математические модели препятствий выбирать в виде выпуклых функций, то локальные минимумы практически исключаются, но увеличивается число возможных маршрутов, в результате чего увеличивается время выполнения алгоритма. Кроме того, матрицы смежности традиционного муравьиного алгоритма являются слишком большими, в результате чего значительно снижается скорость алгоритма.

Улучшения традиционного муравьиного алгоритма включают:

– Построение новой функции вероятности перехода, позволяющей уменьшить рассматриваемое число путей:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ig}^\beta \theta_{ij}^\gamma d_{ij}^\varepsilon}{\sum \tau_{ij}^\alpha \eta_{ig}^\beta \theta_{ij}^\gamma d_{ij}^\varepsilon}, & i \in allowed_k, j \in allowed_k \\ 0, & others, \end{cases} \quad (22)$$

где $\eta_{jg} = 1/d_{jg} = 1/\sqrt{(x_j - x_g)^2 + (y_j - y_g)^2}$ – величина, обратная длине дуги jg ; θ_{ij} – величина, обратная углу $\angle jig$; $d_{ij} = 1/\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ – величина, обратная длине дуги ij ; γ, ε – параметр влияния.

– Предлагается также устранить принятое в традиционном алгоритме ограничение длины поиска на каждом шаге ближайшими узлами сетки. Если разрешить переход от текущей точки к любой точке, которые не находятся в запрещенной области количество проходимых точек поиска значительно снижается, что позволяет значительно повысить скорость расчёта алгоритма планирования.

– Для уменьшения размера матрицы смежности предлагается при вычислении вероятности перехода определять следующую точку перехода и рассчитывать матрицы смежности только для проходимых точек. Если в карте поиска существуют M точек поиска и среди них N точек находится за пределами препятствий, то размер матрицы смежности можно записать следующим образом:

$$Q_T = M \times M; Q_Y = \sum_{i=1}^k \sum_{j=1}^m n_{ij} \times (M - N) \quad (23)$$

где Q_T – размер матрицы смежности традиционного алгоритма; Q_Y – улучшенного алгоритма; n_{ij} – количество проходимых точек j -ого муравья в i -ой итерации; k – количество итерации; m – количество муравьев. В общем случае $Q_Y \ll Q_T$, что позволяет ускорить вычисления.

Определим область (сферу) наблюдения перед препятствием и запрещенную область перехода для каждой точки:

$$Scope_P_j = \sum_{i=1}^{obs-num} P_j_scopei, P_j_fob = \sum_{i=1}^{obs-num} P_j_fobi \quad (24)$$

где $obs-num$ – количество препятствий; P_j – точка маршрута. Обозначим буквой O – начальную точку, буквой G – целевую.

Алгоритм избегания препятствий в режиме реального времени состоит в следующем:

– Если вектор OG не в сфере наблюдения перед препятствием $Scope_O$, то не надо создавать облачно-точечную карту, так как робот может двигаться непосредственно от текущей позиции в целевую точку, как показано на рис. 11а.

– Если вектор P_jG входит в сферу $Scope_P_j$, то робот движется от текущей позиции в точку P_{j+1} , которая имеет наибольшую вероятность перехода и не находится в области P_j_fob .

– Если вектор PG входит в сферу $Scope_P$, то остановить поиск маршрута, то есть робот может двигаться непосредственно из текущей позиции в целевую точку, как показано на рис. 7 б для точки P_2 .

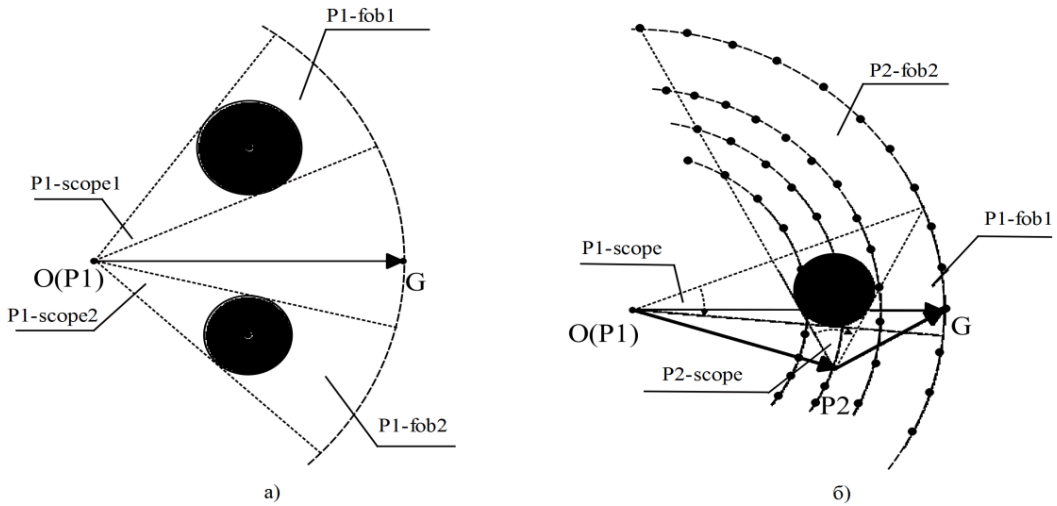


Рис. 7. Стратегия избегания препятствий

Работоспособность алгоритма проверялась моделированием планирования маршрута в среде Matlab. В области моделирования существуют четыре препятствия с разными радиусами; G – целевая точка; O – начальная точка. Чтобы проиллюстрировать преимущества предлагаемого улучшенного муравьиного алгоритма (УМА), сравним результаты планирования маршрута с традиционным муравьиным алгоритмом (ТМА). Результаты моделирования показаны на рис. 8.

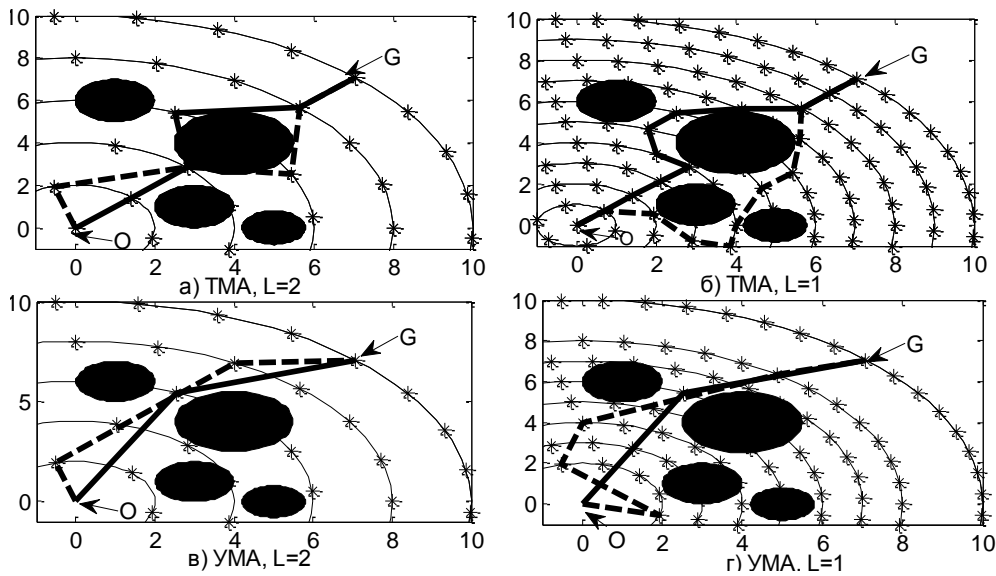


Рис. 8. Результаты моделирования планирования маршрута

На рис. 8 точки со звездами – точки поиска маршрута; рисунки рис. 8а и 8в – результаты при расстоянии между кругами «сетки» $L = 2$; рис. 8б и 8г – результаты при расстоянии между кругами «сетки» $L = 1$; толстые сплошные линии – результаты планирования маршрута при числе итераций $N = 20$; толстые пунктирные линии – результаты планирования маршрута при числе итераций $N = 3$. Количественные результаты приведены в таблице.

Сравнение результатов моделирования показывает, что предложенный алгоритм работоспособен и работает быстрее, чем традиционный алгоритм; спланированный маршрут обходит все препятствия; длина маршрута и время вычисления зависят от числа итераций, количества точек маршрута поиска и числа шагов поиска.

Блок отслеживания запланированного маршрута. В данной статье предлагается алгоритм генерирования нормального ускорения для отслеживания запланированного маршрута в режиме реального времени для квадрокоптера на основе управления нормальным ускорением, вычисленным с помощью скорости полета, угол упреждения и расстояния между положением центра массы квадрокоптера и положением опорной точки (L_1), с адаптивным выбором точки маршрута в качестве опорной точки для упрощения вычисления алгоритма.

Если расстояние между положением центра массы квадрокоптера и положением опорной точки на запланированной точке равно L_1 , как показано на рис. 9:

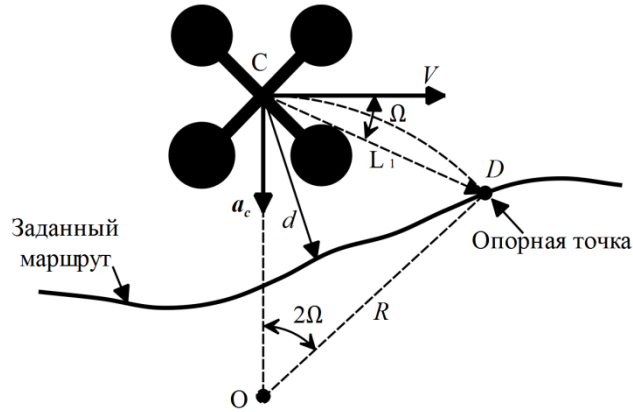


Рис. 9. Следование по запланированному маршруту квадрокоптера

На рисунке знаки: \mathbf{V} — вектор скорости квадрокоптера; \mathbf{L}_1 — вектор от центра массы квадрокоптера до опорной точки; Ω — угол упреждения между векторами \mathbf{V} и \mathbf{L}_1 ; d — кратчайшее расстояние от центра массы квадрокоптера до опорной точки; \mathbf{a}_c — вектор нормального ускорения для управления полетом квадрокоптера; D — опорная точка на запланированном маршруте; O — центр вспомогательной окружности; R — радиус вспомогательной окружности; C — точка центра массы квадрокоптера.

То нормальное ускорение полета квадрокоптера можно определить следующим образом:

$$a_c = \frac{V^2}{R} = \frac{V^2}{L_1 / (2 \sin \Omega)} = \frac{2V^2}{L_1} \sin \Omega \quad (25)$$

На формуле (25) видно, что необходимая разработка для определения нормального ускорения управления полетом отслеживания запланированного маршрута является определением вектора \mathbf{L}_1 , т. е. выбором опорной точки.

Расстояние L_1 является критическим параметром для алгоритма отслеживания заданного маршрута. Оно существенно влияет на кривую траектории движения квадрокоптера. Если величина расстояния большая, квадрокоптер летит по дуге с меньшей кривизной и медленно приближается к заданному маршруту. В этом случае не происходит больших колебаний траектории полета квадрокоптера, но реальная траектория полета может иметь большее отклонение от заданного маршрута. Если расстояние имеет меньшее значение, квадрокоптер летит по дуге с большей кривизной и быстро приближается к заданному маршруту, но будет производить большие колебания траектории полета квадрокоптера, реальная траектория полета тоже может быть иметь большее отклонение от заданного маршрута.

Для решения этого вопроса, в данной статье предлагается алгоритм отслеживания заданного маршрута на основе адаптивного выбора вектора, т. е. адаптивного выбора опорной точки в связи с вектором скорости полета квадрокоптера и характеристиками заданного маршрута.

Процесс адаптивного выбора опорной точки может разделить на два этапа:

– выбор ближайшей точки маршрута D_1 на заданном маршруте от текущего положения центра массы квадрокоптера C под условия: $L_1 \geq R_C$; $\Omega_1 \leq \pi/2$.

– определение номера выбранной опорной точки можно записать следующим образом:

$$N = n + \phi \quad (26)$$

где N - серийный номер опорной точки в множестве точек маршрута; n - серийный номер точки D_1 в множестве точек маршрута; ϕ - функция выбора опорной точки, которую может записать следующим образом:

$$\phi = \begin{cases} 0, & m = 1 \\ \left\lfloor \frac{m}{\left[(1 + \sin \Omega_1)^p (1 + d)^q \left(1 + \sum_{i=1}^{m-1} (\pi - \chi_i) \right)^r \right]} \right\rfloor, & m \geq 2 \end{cases} \quad (27)$$

Где p, q, r - коэффициент фактора влияния; m - количество точек в окружности вокруг точки D_1 с радиусом R_T , которое может отражать плотность точки в этой области, с другой стороны, тоже можно

отражать распределение препятствий; χ - угол между точками маршрута, который в определенной степени отражает кривизну заданного маршрута в области окружности вокруг точки D_1 с радиусом R_T ;

$\sum_i^{m-1} (\pi - \chi_i)$ - отражает изменения кривизны заданной траектории в определенной степени.

Работоспособность нашего алгоритма отслеживания маршрута проверялась сравнением результатов моделирования отслеживания трехмерной прямой линии и трехмерной спиральной линии при адаптивном выборе точки маршрута в качестве опорной точки и выборе каждой точки в качестве опорной точки в множестве точки маршрута в среде Matlab. Результаты моделирования алгоритма показаны на рис. 10.

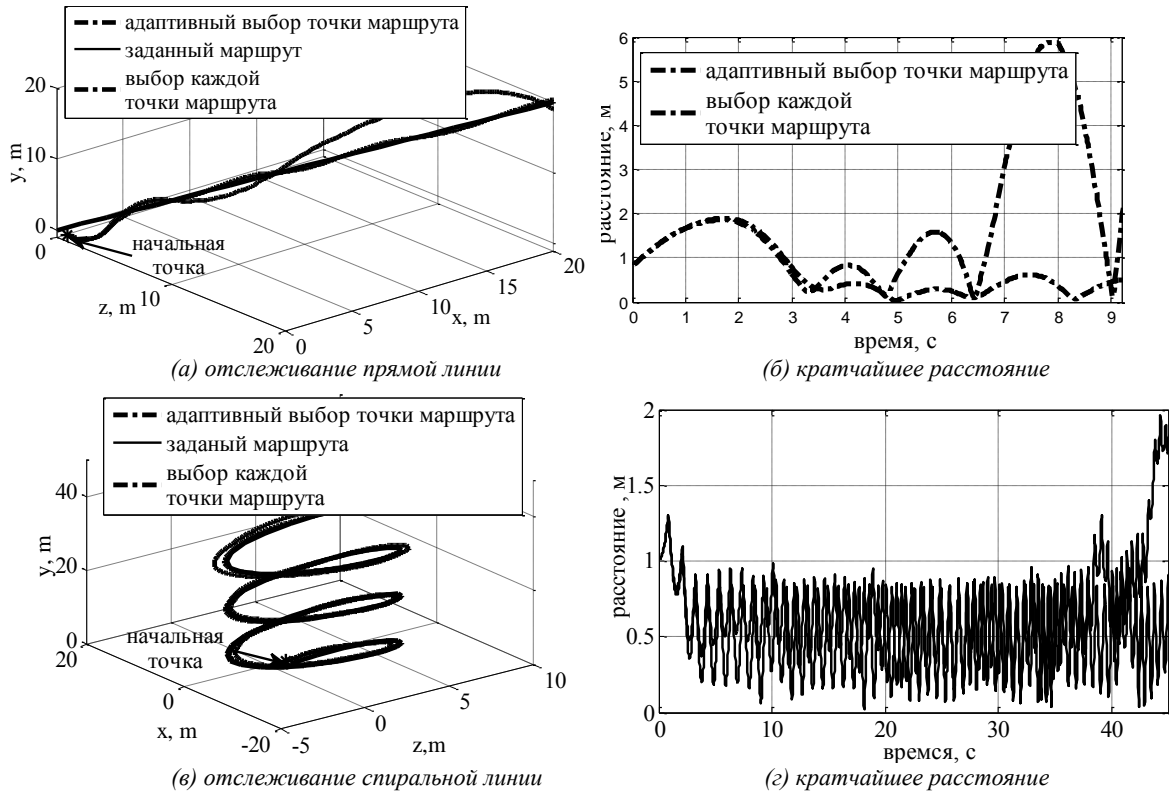


Рис. 10. Результаты отслеживания трехмерных маршрутов

Из результатов моделирования, показанных на рисунке 16, можно заметить, что наш алгоритм может быть использован для отслеживания трехмерного маршрута, даже еще более сложной трехмерной спиральной линии. Кроме того, если квадрокоптер летит под управлением алгоритма отслеживания с выбором каждой точки, угол упреждения слишком большой, это будет влиять на скорость сходимости, и даже расходятся.

Блок облета препятствий. В процессе полета по заданному маршруту, квадрокоптеру необходимо иметь возможность автономного облета препятствий, включая заранее неизвестные, подвижные, и заранее известные. В этой статье предлагается онлайн-метод обхода неподвижных и подвижных препятствий с неизвестными скоростями и ускорениями в динамической среде в реальном времени, на основе управления боковым ускорением. Сначала получаем информацию о состоянии движения робота и информацию о препятствиях предыдущего момента и предсказать состояния движения препятствий и робота следующего момента. Если в среде отсутствуют препятствия, которые не соответствуют условиям обхода, то вычислить минимальное ускорение сближения к целевой точке. Если ситуация противоположная, то вычислить минимальное ускорение обхода препятствий. Если в среде существуют подвижные препятствия, то предсказать позицию столкновения при расчете экстремум функции расстояния и положить виртуальные статические препятствия в этом месте. Умножить минимальное ускорение обхода на коэффициент обхода, чтобы более безопасно проходить мимо препятствий. Умножить минимальное ускорение сближения на коэффициент сближения, чтобы более быстро двигаться к целевой точке. Ограничить скорость движения, чтобы боковое ускорение было не выше чем максимальное безопасное ускорение робота.

На практике, системы работы часто являются линейными, не может удовлетворить условиям использования стандартного фильтра Калмана. Для нелинейных систем обычно используем расширенный фильтр Калмана (ЕКФ). Но его недостатки очевидны: необходимая линеаризация для нелинейной

системы, обычно с помощью разложения ряда Тейлора, сохраняя член первого порядка и игнорируя члены высшего порядка, которые может привести к большим ошибкам к сильно-нелинейной системе даже расхождение; в процессе фильтрации необходимо вычисление матрицы якоби, но для нелинейной функции трудно её получить, и значительно уменьшить скорость вычислений. Чтобы преодолеть эти недостатки Julier и др. предложили без запаха фильтр Калмана (UKF) фильтр на основе преобразования без запаха преобразование (UT), который является другим видом широко используемого нелинейного фильтра [26-27].

Для записи препятствий, нужно создать модель препятствий, которые могут иметь разные формы и размеры. Для обхода препятствий важно эффективно описать препятствия. Моделирование препятствий полигонами неудобно для математических вычислений. Для упрощения расчета и моделирования предлагается в качестве модели любых препятствий брать круги, описывающие реальную форму соответствующих препятствий. Такие модели препятствий показаны на рис. 11. Нерегулярные препятствия заменены описанными кругами. Относительные отношения положения и углов являются важной информацией для выполнения обхода столкновения с препятствием, как показано на рис. 11, где \vec{V} - вектор скорости робота; C - центр масса робота; два пунктирные линии представляют касательные линии от точки к кругу; r_b - радиус круга; B - центр радиуса; α - угол между касательными линиями; θ - угол между вектором скорости и вектором \vec{CB} ; E_1, E_2 - касательные точки круга; d - расстояние между центром массы робота и центром круга.

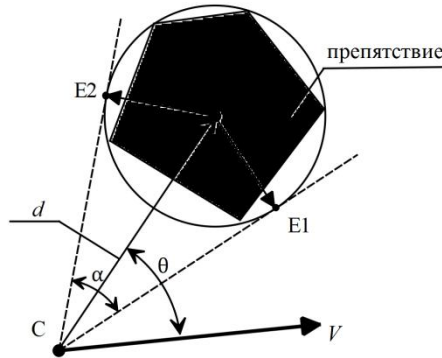


Рис. 11. Модель препятствия и относительное отношение между роботом и препятствием

Вектор состояния движения робота и препятствий в моменте $k + 1$:

$$X_{k+1} = [X_{b_{k+1}}, V_{b_{k+1}}, R_{k+1}]^T; \quad (28)$$

где $X_{b_{k+1}} = [x_{b1_{k+1}}, y_{b1_{k+1}}, x_{b2_{k+1}}, y_{b2_{k+1}}, \dots, x_{bi_{k+1}}, y_{bi_{k+1}}, \dots, x_{bn_{k+1}}, y_{bn_{k+1}}]$ - координаты препятствий, n - количество препятствий, соответствующих условия обхода; $V_{b_{k+1}} = [v_{bx1_{k+1}}, v_{by1_{k+1}}, v_{bx2_{k+1}}, v_{by2_{k+1}}, \dots, v_{bxi_{k+1}}, v_{byi_{k+1}}, \dots, v_{bxn_{k+1}}, v_{byn_{k+1}}]$ - скорость препятствий; $R_{k+1} = [r_{1_{k+1}}, r_{2_{k+1}}, \dots, r_{i_{k+1}}, \dots, r_{n_{k+1}}]$

Если время выборки очень короткое, то можно предположить в течение этого времени ускорения препятствий и роботов не изменены. Можно получить предсказанные состояния вектора следующего момента из состояния вектора текущего момента:

$$\begin{cases} x_{bi_{k+1}} = x_{bi_k} + v_{bxi_k} \cdot \Delta t + 1/2 \cdot a_{bxi_k} \cdot \Delta t^2 \\ y_{bi_{k+1}} = y_{bi_k} + v_{byi_k} \cdot \Delta t + 1/2 \cdot a_{byi_k} \cdot \Delta t^2 \\ v_{bxi_{k+1}} = v_{bxi_k} + a_{bxi_k} \cdot \Delta t \\ v_{byi_{k+1}} = v_{byi_k} + a_{byi_k} \cdot \Delta t \\ r_{b_{k+1}} = r_{b_k} \end{cases}, i = 1, 2, \dots, n \quad (29)$$

Вектор наблюдения в моменте k :

$$Z_k = [D_k, \alpha_k, \theta_k]^T \quad (30)$$

где $D_k = [d_{1_k}, d_{2_k}, \dots, d_{i_k}, \dots, d_{n_k}]$ - вектор расстояния; $\alpha_k = [\alpha_{1_k}, \alpha_{2_k}, \dots, \alpha_{i_k}, \dots, \alpha_{n_k}]$ - вектор углов между касательными линиями; $\theta_k = [\theta_{1_k}, \theta_{2_k}, \dots, \theta_{i_k}, \dots, \theta_{n_k}]$ - вектор углов между вектором скорости и вектором \vec{CB} ;

Отношения вектора состояния и вектора наблюдения описаны следующим образом:

$$\begin{cases} d_{i_k} = \sqrt{(x_{c_k} - x_{bi_k})^2 + (y_{bi_k} - y_{c_k})^2} \\ \alpha_{i_k} = 2 \cdot \arcsin \left(\frac{r_{i_k}}{\sqrt{(x_{bi_k} - x_{c_k})^2 + (y_{bi_k} - y_{c_k})^2}} \right) \\ \theta_k = \arccos \left(\frac{(x_{bi_k} - x_{c_k}) \cdot v_x + (y_{bi_k} - y_{c_k}) \cdot v_y}{\sqrt{(x_{bi_k} - x_{c_k})^2 + (y_{bi_k} - y_{c_k})^2} \cdot \sqrt{v_x^2 + v_y^2}} \right) \end{cases}, i=1, 2, \dots, \quad (31)$$

где $[x_{c_k}, y_{c_k}]$ и $[v_x, v_y]$ - координаты и скорости робота k-го момента.

Нелинейная модель квадрокоптера и карты можно записать в следующем виде:

$$\begin{cases} X_{k+1} = f(X_k, u_k) + \varepsilon_k \\ Z_k = h(X_k) + \eta_k \end{cases} \quad (32)$$

где u_k - контрольный сигнал; ε_k, η_k - шум процесса и шум наблюдения с 0 среднего значения и их ковариации Q and R .

Процесс фильтрации Кальмана без запаха может быть разделен на четыре процесса:

Инициализация:

$$\begin{cases} \hat{X}_0 = E[X_0] \\ P_0 = E[(X_0 - \hat{X}_0)(X_0 - \hat{X}_0)^T] \\ \hat{X}_0^a = E[X_0^a] = E[X_0^x, X_0^\varepsilon, X_0^\eta] = [\hat{X}_0^T, 0_{n \times 1}, 0_{m \times 1}]^T \\ P_0^a = E[(X_0^a - \hat{X}_0^a)(X_0^a - \hat{X}_0^a)^T] = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix} \end{cases}$$

где n - размер шума процесса; m - размер шума наблюдения; X_0^x - вектор состояния системы; X_0^ε - вектор состояния шума процесса; X_0^η - вектор состояния шума наблюдения.

Рассчитать Sigma точки:

$$\chi_k^a = [\chi_k^x, \chi_k^\varepsilon, \chi_k^\eta] = [\hat{X}_{k-1}^a, \hat{X}_{k-1}^a + \sqrt{(L+\lambda)P_{k-1}^a}, \hat{X}_{k-1}^a - \sqrt{(L+\lambda)P_{k-1}^a}], L = N + n + m$$

Обновление процесса и Обновление наблюдения:

$$\begin{cases} \chi_{k+1}^x = f[\chi_k^x, \chi_k^\varepsilon] \\ \hat{X}_{k+1}^- = \sum_{i=0}^L W_i^m \chi_{i,k+1}^x \\ P_{k+1}^- = \sum_{i=0}^L W_i^c [\chi_{i,k+1}^x - \hat{X}_{k+1}^-] [\chi_{i,k+1}^x - \hat{X}_{k+1}^-]^T \\ Y_{k+1} = h[\chi_k^x, \chi_k^\eta] \\ \hat{Z}_{k+1}^- = \sum_{i=0}^L W_i^m Y_{i,k+1} \end{cases} \begin{cases} P_{Z_{k+1}Z_{k+1}} = \sum_{i=0}^{2L} W_i^c [Y_{i,k+1} - \hat{Z}_{k+1}^-] [Y_{i,k+1} - \hat{Z}_{k+1}^-]^T \\ P_{X_{k+1}Z_{k+1}} = \sum_{i=0}^{2L} W_i^c [\chi_{i,k+1}^x - \hat{X}_{k+1}^-] [Y_{i,k+1} - \hat{Z}_{k+1}^-]^T \\ K = P_{X_{k+1}Z_{k+1}} P_{Z_{k+1}Z_{k+1}}^{-1} \\ X_{k+1} = X_{k+1}^- + K(Z_{k+1} - \hat{Z}_{k+1}^-) \\ P_{k+1} = P_{k+1}^- - K P_{Z_{k+1}Z_{k+1}} K^T \end{cases}$$

Уравнение бокового ускорения можно записать следующим образом:

$$a_c = k_b \cdot \sum_j^n a_{b_j} \quad (33)$$

где k_b - коэффициент обхода; a_{b_j} - минимальное ускорение обхода; n - количество препятствий.

Для включения процесса обхода неподвижных препятствий, необходимо соответствует условия обхода: $d_i \leq d_r$, d_r - порог расстояния обхода; вектор скорости робота \vec{V}_c находится в секторной области $score_i$ покрывается угол α_i . Если в среде существуют препятствия, которые не соответствуют

условиям обхода, то включить процесс сближения к целевой точкой. На рис. 12 показан случай отсутствия препятствий между квадрокоптером и целевой точкой, расстояние от текущей позиции робота до первого препятствия d_1 более чем порог расстояния обхода d_r , вектор скорости V_c не находится в области $score2$, поэтому робот в процессе сближения с целевой точкой.

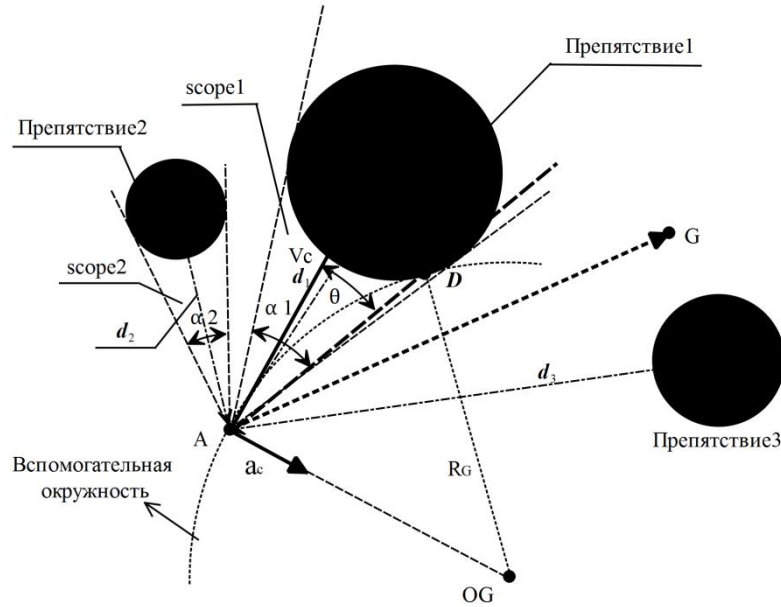


Рис. 12. Процесс облета препятствия

В данном случае минимальное ускорение сближения можно записать следующим образом:

$$a_b = \frac{V_c^2}{R_G} = \frac{V_c^2}{d_{AD}/(2 \cdot \sin \theta)} = \frac{2 \cdot V_c^2 \cdot \sin \theta}{d_{AD}} \quad (34)$$

где d_{AG} - расстояние от текущей точки до целевой точки.

В неизвестной среде робота не только могут избежать столкновения с неподвижными препятствиями, и также должны иметь возможность обхода подвижных препятствий. Робот, обсуждаемый в этой статье, в основном получает информацию о препятствиях с нескольких бортовых датчиков расстояния (см. рис. 13). В реальном пространстве работы препятствия двигаются неравномерно, и трудно получить информацию о движении препятствий точно, в данной статье мы используем фильтр Калмана для уменьшения влияния погрешности измерений и наблюдений. Робот будет обнаруживать препятствия в режиме реального времени и получать информацию препятствий, а затем рассчитать минимальное значение функции расстояния между роботом и препятствием, которую можно записать в следующем образом:

$$\begin{cases} f_j(t) = \sqrt{(x_c - x_{bj})^2 + (y_c - y_{bj})^2} \\ x_c = x_{ck} + v_{cxk} + 0.5 \cdot a_{cxk} \cdot t^2 \\ y_c = y_{ck} + v_{cyk} + 0.5 \cdot a_{cyk} \cdot t^2 \\ x_{bj} = x_{bjk} + v_{bxjk} + 0.5 \cdot a_{bxjk} \cdot t^2 \\ y_{bj} = y_{bjk} + v_{byjk} + 0.5 \cdot a_{byjk} \cdot t^2 \end{cases}, j = [1, 2, \dots, n] \quad (35)$$

где n - количество подвижных препятствий; $[x_{ck}, y_{ck}]$, $[v_{cxk}, v_{cyk}]$ и $[a_{cxk}, a_{cyk}]$ - координаты, скорости и ускорения робота текущего момента; $[x_{bjk}, y_{bjk}]$, $[v_{bxjk}, v_{byjk}]$ и $[a_{bxjk}, a_{byjk}]$ - координаты скорости и ускорения j -го подвижного препятствия текущего момента.

Если минимальное значение функции расстояния $f_j(t_m) = \min f_j(t)$, т. е. $f_j'(t_m) = 0$, то предсказанная позиция столкновения: $[x_c(t_m), y_c(t_m)]$.

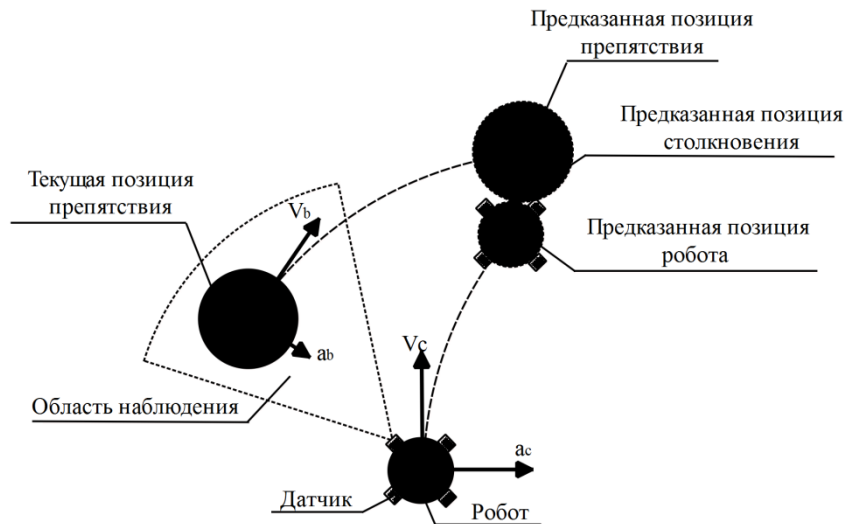


Рис. 13. Модель предсказанного столкновения робота с неподвижным препятствием

После предсказания позиции столкновения поместим виртуальное статическое препятствие в данном месте в качестве объекта обхода.

Работоспособность предложенного алгоритма планирования маршрута и метода облета неподвижных препятствий проверялась сравнением результатов моделирования существующих популярных алгоритмов локального планирования маршрута в среде Matlab. На рис. 14 показаны результаты моделирования маршрута в статической среде.

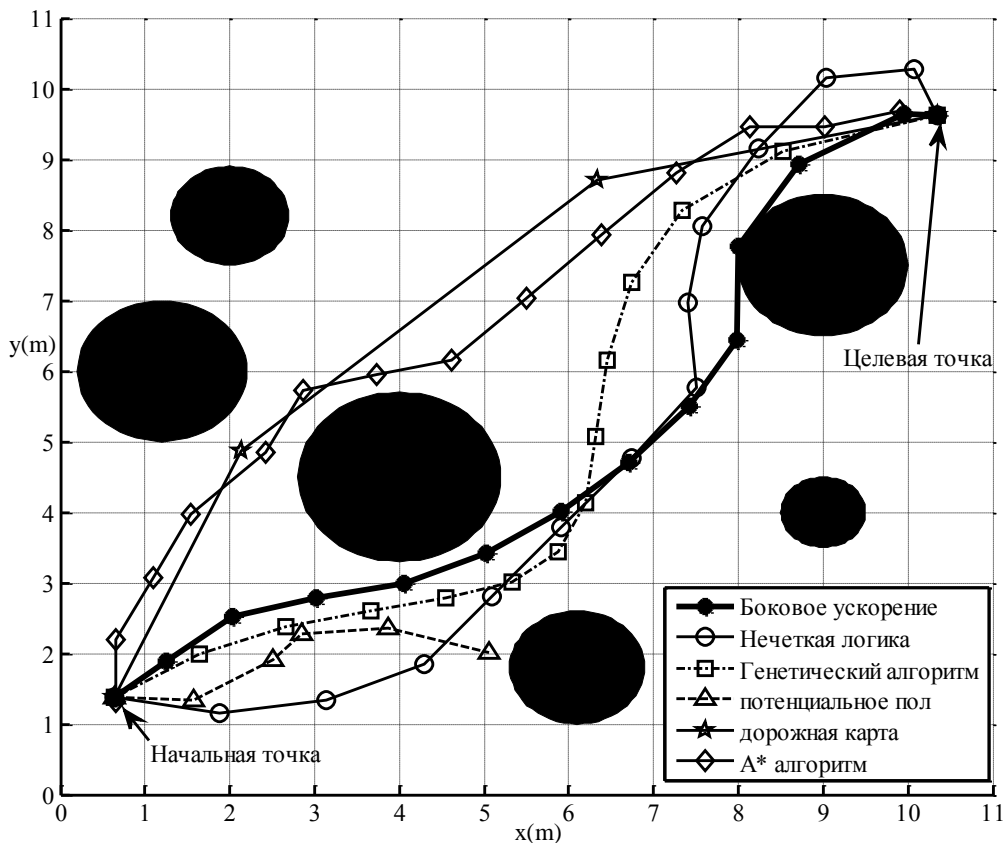


Рис. 14. Планирование маршрута в статической среде

Сравнение результатов моделирования планирования маршрута и обхода неподвижных препятствий приведено в таблице:

Таблица 1. Сравнение результатов

Алгоритм	НЛ	ГЛ	ПП	ДК	A*	БУ
----------	----	----	----	----	----	----

Время(с)	0.7625	2.8273	0.3833	2.1221	2.5506	0.4618
Длина маршрута(м)	18.8074	16.5339	5.8807	17.6715	18.1203	16.2461
Гладкий маршрут	да	да	да	нет	нет	да
Столкновение	нет	нет	да	нет	нет	нет

Результаты моделирования показывают: предложенный алгоритм имеет быстрые скорости вычисления.

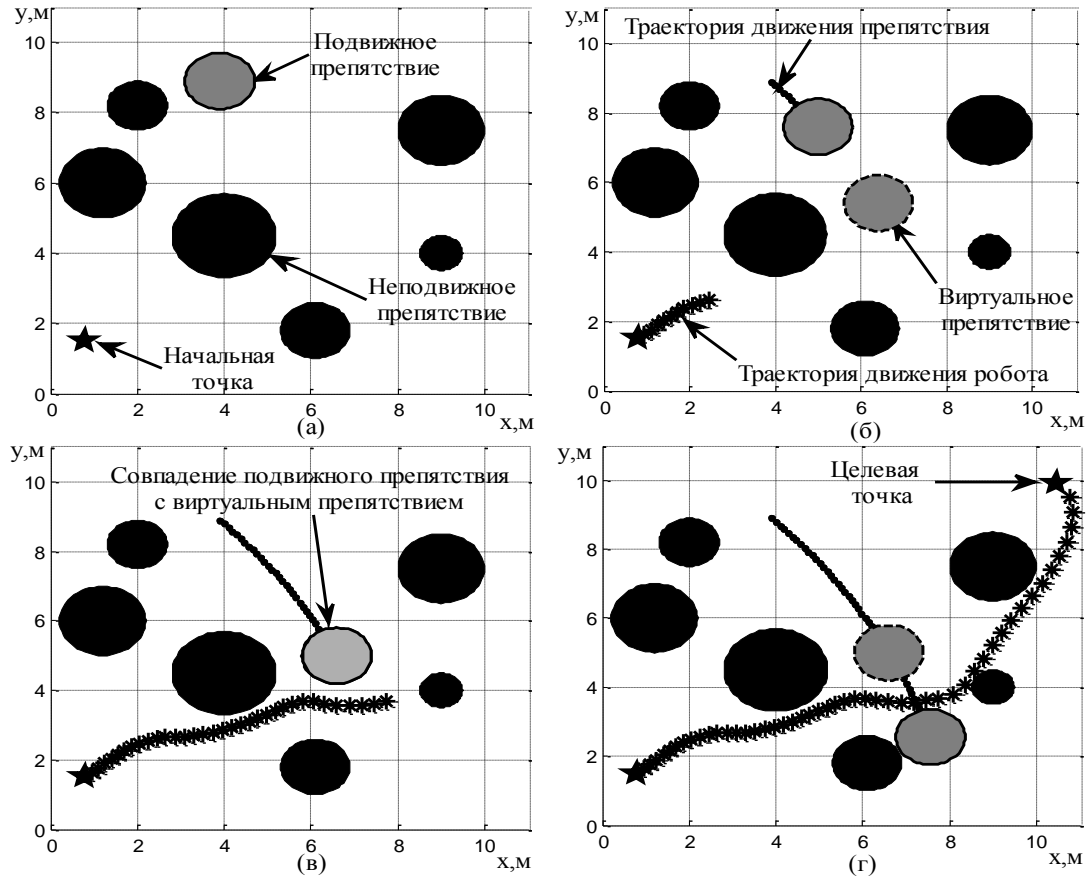
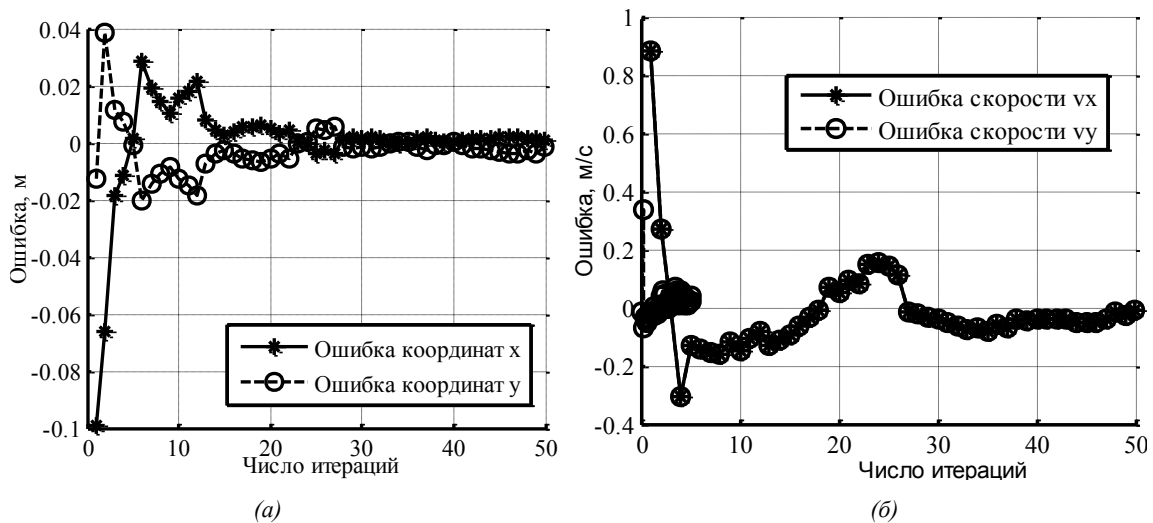
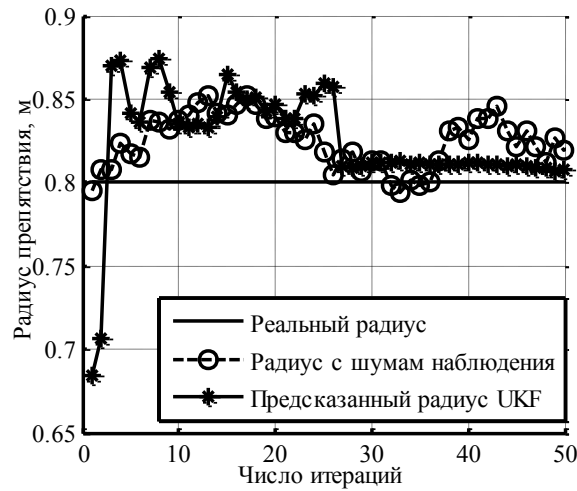


Рис. 15. Планирование маршрута в динамической среде: (а) - начальный момент; (б) - момент обнаружения возможности столкновения; (в) - момент совпадения подвижного препятствия с виртуальным препятствием; (г) - момент прибытия в конечную точку

Результаты фильтрации состояний подвижного препятствия показаны на рис. 16.





(в)

Рис. 16. Фильтрации состояний движения подвижного препятствия: (а) - изменение ошибки координаты, (б) - изменение ошибки скорости, (в) - изменение радиуса

Результаты моделирования показывают, что робот может эффективно избегать столкновения с неподвижными и подвижными препятствиями; движения и радиуса подвижного препятствия.

Моделирование планирования маршрута в сложной динамической среде. Работоспособности алгоритмов проверяются моделированием в сложной динамической среде с учётом влияния ветра, в которой составляются известные препятствия, неизвестные препятствия и подвижные препятствия. Моделирование проведено в программной среде Matlab.

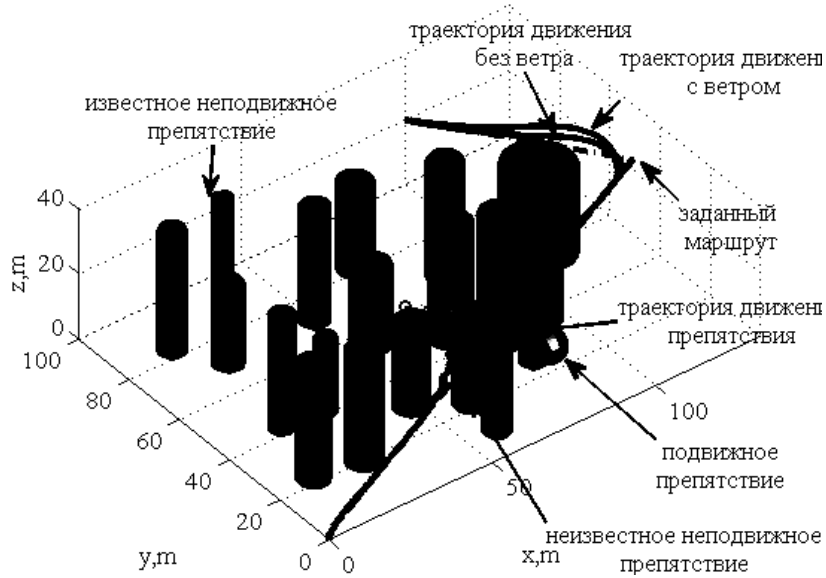
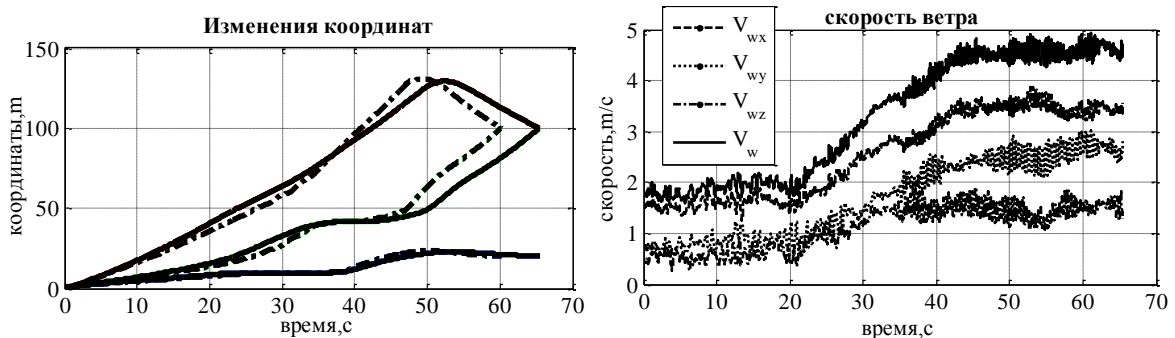


Рис. 17. Траектория полета квадрокоптера

На рис. 18 показаны параметры движения квадрокоптера и скорость ветра.



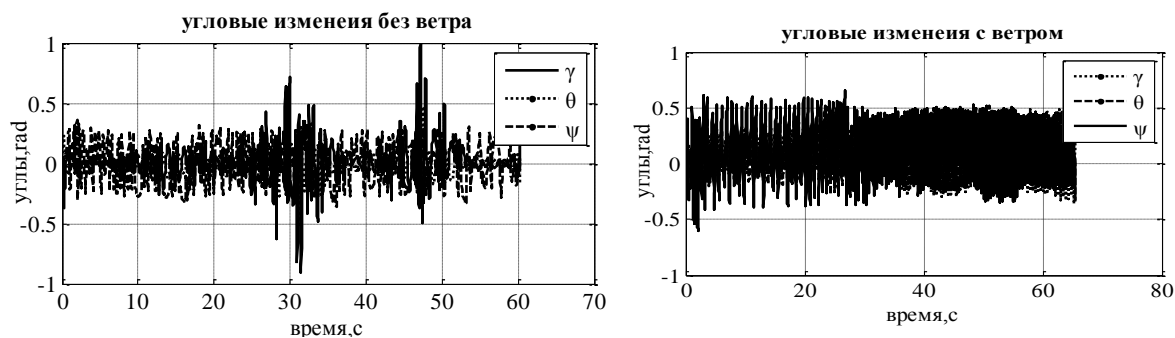


Рис. 18. Параметры движения квадрокоптера и скорость ветра

Из результатов, показанных на рисунке 17 и 18, можно заметить, что траектория полета квадрокоптера избегает столкновения с всеми неподвижными препятствиями и подвижным препятствием, и эта траектория находится под траекторией движения подвижного препятствия.

Заключение. Разработана математическая модель квадрокоптера как объекта управления. Представлена разработка алгоритма управления, который позволяет стабилизировать высоту, угловое положение и координаты полета квадрокоптера и отслеживать заданную траекторию. Разработана многорежимного контроллера, позволяющего работать квадрокоптеру в разных режимах. Построена облачно-точечная карта поиска маршрута; улучшен муравьиный алгоритм путем введения новой функции вероятности перехода, снижения размера матрицы смежности и устранения ограничения длины поиска; представлен метод избегания препятствий в режиме реального времени. Предсказаны состояния движения препятствий с помощью фильтра Калмана без запаха(UKF). Упрощен алгоритм отслеживания заданного маршрута при выборе точки маршрута в качестве опорной точки. Представлена разработка алгоритма отслеживания заданного маршрута с адаптивным выбором точки маршрута в качестве опорной точки. Представлен новый онлайн-метод обхода неподвижных и подвижных препятствий в режиме реального времени на основе управления боковым ускорением. Результаты моделирования показывают хорошую работоспособность для обхода неподвижных препятствий в динамической среде.

Литература

1. Omar Santos-Hugo Romero, Sergio Salazar-Rogelio Lozano. Real-time Stabilization of a Quadrotor UAV: Nonlinear Optimal and Suboptimal Control, J Intell Robot Syst., 2013. № 70: P. 79–91. DOI: 10.1007/s10846-012-9711-8.
2. Tong Li, Youmin Zhang and Brandon W. Gordon. Passive and active nonlinear fault tolerant control of a quadrotor unmanned aerial vehicle based on the sliding mode control technique. Journal of Systems and Control Engineering, 2013. № 227. P. 12–23. DOI: 10.1177/0959651812455293.
3. Yue Bai, Cheng Peng, Changjun Zhao, Yantao Tian. Trajectory tracking control of a quad-rotor UAV based on command filtered backstepping. // 2012 Third International Conference on Intelligent Control and Information Processing. IEEE, 2012. P. 179-184. DOI:10.1109/ICICIP.2012.6391413.
4. Dierks T. and Jagannathan S. Output feedback control of a quadrotor UAV using neural networks. IEEE Trans. Neural Netw., 2010. № 21 (1). P. 50–66.
5. Xian B., Diao C., Zhao B. et al. Nonlinear robust output feedback tracking control of a quadrotor UAV using quaternion representation [J]. Nonlinear Dynamics, 2014. P. 1-18.
6. Yi Z., Xiuxia Y., Hewei Z. et al. Tracking control for UAV trajectory [C] // Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese. IEEE, 2014. P. 1889-1894.
7. Andrew Roberts and Abdelhamid Tayebi. Adaptive Position Tracking of vtol uavs. iee trans. on rob. Vol. 27. № 1, February (2011).
8. Barraquand J., Latombe J. C. Robot motion planning: A distributed representation approach // The International Journal of Robotics Research, 1991. № 10 (6). P. 628-649.
9. Bhattacharya P., Gavrilova M. L. Roadmap-based path planning-Using the Voronoi diagram for a clearance-based shortest path // Robotics & Automation Magazine, IEEE, 2008. № 15 (2). P. 58-66.
10. Гаспилович Е. А., Дементьев В. А., Мишенина К. А. Потенциальное поле и частоты колебаний молекулы антрахинона // Журнал физической химии, 1981. С. 55-73.
11. Лебедев О. Б., Лебедева Е. М., Пестов В. А. Алгоритмы планирования движения подводного аппарата // Известия Южного федерального университета. Технические науки, 2015. № (165). С. 67-77.
12. Tkachev S. B., Liu W. Design of Path Following Method for Unmanned Aerial Vehicles using Normal Forms // IFAC-PapersOnLine, 2015. № 48 (11). P. 10-15.
13. Aguiar A. P., Hespanha J. P. Trajectory-tracking and path-following of underactuated autonomous vehicles

- with parametric modeling uncertainty// IEEE Transactions on Automatic Control, 2007. № 52 (8). C. 1362-1379.
14. *Zuo Z.* Trajectory tracking control design with command-filtered compensation for a quadrotor // Control Theory & Applications, IET, 2010. № 4 (11). C. 2343-2355.
 15. *Takeshi A.* Motion planning for multiple obstacles avoidance of autonomous mobile robot using hierarchical fuzzy rule[C]//Proceedings of IEEE International Conference on Multi-sensor Fusion and Integration for Intelligent System (MFI'94). Las Vegas, 1994. P. 265-271.
 16. *Mbede J. B., Huang X., Wang M.* Fuzzy motion planning among dynamic obstacles using artificial potential fields for robot manipulators [J]. Robotics and autonomous Systems, 2000. № 32 (1). P. 61-72.
 17. *Jaradat M. A. K., Garibeh M. H., Feilat E. A.* Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field[J]. Soft Computing, 2012. № 16 (1). P. 153-164.
 18. *Yang S. X., Meng M.* An efficient neural network approach to dynamic robot motion planning [J]. Neural Networks, 2000. № 13 (2). P. 143-148.
 19. *Yang S. X., Meng M.* Neural network approaches to dynamic collision-free trajectory generation[J]. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2001. № 31 (3). P. 302-318.
 20. *Chungang Z., Yugeng X.* Robot path planning in globally unknown environments based on rolling windows[J]. Science in China Series E: Technological Science, 2001. № 44 (2). P. 131-139.
 21. *McCormick B. W.* Aerodynamics of V/STOL flight [M]. Courier Corporation, 1967.
 22. *Munoz L. E., Santos O., Castillo P.* Robust nonlinear real-time control strategy to stabilize a PVTOL aircraft in crosswind[C] // Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, 2010. P. 1606-1611.
 23. *Bangura M., Mahony R.* Nonlinear dynamic modeling for high performance control of a quadrotor [C] // Australasian conference on robotics and automation, 2012. P. 1-10. Belatti T. quadrotor flight in constrained environments [j].
 24. *Moyano Cano J.* Quadrotor UAV for wind profile characterization [J], 2013.
 25. *Bouabdallah S., Siegwart R.* Backstepping and sliding-mode techniques applied to an indoor micro quadrotor[C] // Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. IEEE, 2005. P. 2247-2252.
 26. *Julier S. J., Uhlmann J. K.* Unscented filtering and nonlinear estimation [J]. Proceedings of the IEEE, 2004. № 92(3). P. 401-422.
 27. *Wan E., Van der Merwe R.* The unscented Kalman filter for nonlinear estimation [C] //Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE, 2000. P. 153-158.