

М. СЛИМ, М. САЙЕД, Х. МАЗЕХ, Х. ШРАИМ, К. ФРАНСИС

## ОТКАЗОУСТОЙЧИВОЕ УПРАВЛЕНИЕ ГРУППОВЫМ ПОЛЕТОМ МУЛЬТИКОПТЕРОВ

*В статье для решения задачи оперативной оптимизации в процессе распределенного управления строем гексакоптеров без лидера используется алгоритм управления искусственным пчелиным роем, наилучший в глобальном смысле (Gbest-Guided Artificial Bee Colony – GABC, АУПР-ГЛ). АУПР-ГЛ обеспечивает оптимизацию целевой функции для каждого агента при приближении гексакоптера к заданной точке и обходе препятствий и других беспилотных летательных аппаратов (БПЛА). Показано, что он может конкурировать с другими применяемыми бионическими алгоритмами, например оптимизации роя частиц (Particle Swarm Optimization – PSO, ОРЧ). Представлены методы отказоустойчивого управления, которые протестированы на разных сценариях, в частности для случая потери агентов и отказа актюаторов в строе. Результаты демонстрируют, что предлагаемые методы отказоустойчивого управления способны минимизировать влияние отказов на конечную задачу строя БПЛА.*

**Ключевые слова:** беспилотные летательные аппараты, алгоритм управления пчелиным роем, управление строем БПЛА, отказоустойчивое управление.

### 1. Введение

В последнее время значительное развитие получила технология создания относительно недорогих винтокрылых БПЛА, оборудованных актюаторами и измерительными датчиками. Благодаря растущим возможностям новых технологий: беспроводной связи и устройств для вычисления и хранения данных – эти аппараты стали способны осуществлять связь на ближних расстояниях, а также обеспечивать локальное принятие решений. Соответственно, мультиагентные системы БПЛА получили широкое распространение в различных приложениях и стали предметом множества исследований [1–4]. Например, в [5–7] рассмотрено применение строя квадрокоптеров в комплекте с различными контроллерами для перевозки больших грузов с целью минимизации их повреждения. БПЛА используются также для съемки земельных участков и полей, картографирования и локализации [8]. Задача обследования и мониторинга наружных территорий группой БПЛА описывается в работе [9] и других публикациях.

**Слим** Малак. Аспирант, Американский университет Бейрута (Ливан); Научно-исследовательский центр приборостроения, факультет приборостроения, Ливанский университет (Хадат, Ливан).

**Сайед** Мадж. Доцент, Ливанский университет.

**Мазех** Хусейн. Преподаватель, Университетский колледж Матна (Бейрут, Ливан); Ливанский университет.

**Шраим** Хасан. Декан, Ливанский университет.

**Франсис** Кловис. Директор, Научно-исследовательский центр приборостроения, Ливанский университет.

Научный редактор перевода д.т.н. И. Б. Фуртат.

В литературе рассматриваются три основных подхода к управлению строем БПЛА: на основе структуры типа «ведущий–ведомый» [10, 11], применения виртуальной [12, 13] и поведенческой структур [14–16]. В поведенческом подходе каждый участник соблюдает набор правил, чтобы выполнить заданные цели строя: предотвращение столкновений, движение с заданной скоростью, центрирование строя. Затем эти цели объединяются в одну общую – достижение заданной траектории, известной всем агентам строя, или точки встречи. Основное преимущество данной структуры состоит в ее простой самоорганизации, а также масштабируемости при изменении количества агентов. Помимо трех главных структур, можно выделить три типа архитектуры управления: централизованную [17, 18], распределенную [15, 19] и децентрализованную [20, 21]. В широко применяемой распределенной архитектуре управления каждый БПЛА в строе оснащается бортовым контроллером, который запрашивает информацию от самого БПЛА и его соседей.

Основная трудность при групповом полете состоит в том, чтобы найти алгоритм управления и правила локального взаимодействия, обеспечивающие эффективное достижение глобальной цели. В литературе предлагается использовать алгоритм консенсуса, который позволяет достичь соглашения между всеми агентами мультиагентной системы, чтобы прийти к заданной цели строя [10, 22, 23]. В данном алгоритме применяется теория алгебраических графов для задания информационного обмена между агентами и их соседями. В других подходах используются методики оптимизации для разработки контроллеров на основе критерия оптимизации. Их основное преимущество в контексте разработки, управления и наблюдения за такими структурами состоит в возможности понимания цели как задачи оптимизации. Стратегия распределенного управления строем винтокрылых БПЛА без лидера (ведущей подсистемы) описывается в [24] и [25], где задача формирования строя решается с помощью алгоритма ОРЧ, чтобы усовершенствовать целевую функцию для каждого аппарата. В результате строй БПЛА достигает пункта назначения в нужной компоновке, избегая препятствий и столкновений между аппаратами. Утверждается, что такая методика обеспечивает оптимальные результаты при малом времени вычислений. Тем не менее, согласно [26], алгоритм ОРЧ может сходиться преждевременно и попадать в локальный минимум, что, в свою очередь, влечет за собой проблемы при работе с такими системами, как мультикоптеры. Чтобы избежать подобных рисков, при решении задачи управления строем БПЛА следует внедрять другие методы оптимизации.

Алгоритм управления пчелиным роем (АУПР) – это популяционный метод оптимизации, способный работать с задачами оптимизации с ограничениями [27]. Сравнение данного алгоритма с другими широко применяемыми методиками оптимизации показывает, что он превосходит их или конкурирует с ними на равных. Во многих исследованиях предлагается использовать АУПР для планирования траектории строя БПЛА, с тем чтобы выбрать кратчайшую для каждого аппарата во время облета препятствий [28]. В [29] рассматривается задача управления группой спутников, где коэффициент усиления обратной связи, обеспечивающий связанное движение и минимальный расход топлива, оценивался методом АУПР. В упомянутых публикациях продемонстрирована польза применения данного алгоритма для планирования траектории и настройки коэффициентов усиления контроллеров, его оптимальная работа и высокая скорость сходимости.

Как говорилось ранее, управление строем БПЛА основывается на передовых технологиях, чья эффективность постоянно повышается. Вместе с тем, если в строю обнаруживаются отказы, эти стратегии могут привести к нежелательному поведению аппаратов и даже вызвать их нестабильность. В этом случае необходимо разработать методы отказоустойчивого управления, чтобы система могла выполнять свои функции, несмотря на сбои. Поскольку методы отказоустойчивого управления чаще применяются для одиночных динамических систем, публикаций по их использованию в мультиагентных системах совсем немного. В работе [30] рассматривается задача консенсуса в мультиагентных системах при обнаружении сбоя актюатора. Показано, что влияние отказов можно эффективно скомпенсировать за счет применения метода на основе виртуальных актюаторов. В [31] предлагается метод отказоустойчивого управления с упреждением для стабилизации и управления движением разнородной группы агентов в трехмерном пространстве. Задачи, которые связаны с обнаружением отказов, возникающих при групповом управлении, рассмотрены в [32]. Представлены результаты управления строем квадрокоптеров на основе алгоритма консенсуса и ОРЧ.

В настоящей работе для решения задачи оперативной оптимизации при распределенном управлении групповым полетом шестивинтовых БПЛА (гексакоптеров) без лидера предлагается использовать АУПР-ГЛ, который обеспечивает оптимизацию целевой функции для каждого аппарата при его приближении к заданной точке, облете препятствий и других БПЛА. Затем рассматриваются случаи потери аппаратов вследствие отказов актюаторов в строю. Испытания проводились на строю из шести гексакоптеров путем моделирования. Результаты демонстрируют, что данный алгоритм перспективен для решения задачи управления строем БПЛА, а предлагаемые методы управления позволяют строю достичь цели, несмотря на возникновение сбоев.

Статья организована следующим образом. В разделе 2 ставится задача и приводятся предварительные данные. В разделе 3 описывается АУПР и его модифицированная версия. В разделе 4 рассматривается отказоустойчивость при управлении строем БПЛА. В разделе 5 приводятся основные результаты. В разделе 6 делаются выводы и обсуждаются перспективы разработки отказоустойчивого управления для мультиагентных систем.

## 2. Предварительные данные

Ниже приводятся основные определения и данные, которые будут использоваться для моделирования и анализа.

### 2.1 Теория графов

В данном разделе архитектура системы гексакоптеров представлена с помощью теории графов. Гексакоптеры обладают следующими свойствами: автономностью, локальным представлением (ни один БПЛА не имеет глобального представления о системе), децентрализацией (каждый гексакоптер взаимодействует только с несколькими соседними БПЛА на основе относительных данных от соседей) и эволюцией во времени (состояние каждого гексакоптера меняется в соответствии с определенными локальными протоколами координации, взаимодействующими друг с другом), что в конечном итоге приводит к коллективному поведению всей системы.

Система гексакоптеров представлена неориентированным графом  $G = (V, \mathcal{E})$  [33], где  $V = \{1, 2, \dots, N\}$  – множество вершин и  $\mathcal{E} \subseteq \{(i, j): i, j \in V, i \neq j\}$  – множество ребер. Вершины называются смежными, если они соединены между собой ребрами. Каждый узел ассоциирован с гексакоптером, а ребра – со связями между агентами в каждой паре.

Матрица смежности  $A \in \mathfrak{R}^{N \times N}$  определяется как матрица размерности  $N \times N$  с элементами  $a_{ij} > 0$ , если  $(i, j) \in \mathcal{E}$  или, в противном случае,  $a_{ij} = 0$ . Множество соседних подсистем  $\mathbb{N} = \{j \in V: (i, j) \in \mathcal{E}\}$   $i$ -го гексакоптера обычно состоит из индексов  $j$ -х гексакоптеров, которые взаимодействуют с  $i$ -м БПЛА. Если  $a_{ij} > 0$ , то аппарат  $j$  является соседним для аппарата  $i$ .

## 2.2 Топология строя гексакоптеров

Чтобы описать строй гексакоптеров в евклидовом пространстве, вектор положения  $\xi_i = [x_{1i}, x_{2i}, \dots, x_{fi}]^T \in \mathfrak{R}^f$  связывается с каждым  $i$ -м узлом графа  $G$ , где  $f$  – размерность пространства, в данном случае  $f = 2$ . Тогда вся конфигурация описывается вектором  $\xi = \text{col}(\xi_1, \dots, \xi_n) \in \mathfrak{R}^{f \times N}$ .

Соседние подсистемы для  $i$ -го гексакоптера определяются как

$$\epsilon_i = \{j \in N: |\xi_i - \xi_j| < l\}, \quad (1)$$

где  $l$  – соседняя область,  $|\cdot|$  – евклидова норма.

Заданную конфигурацию агентов в строе можно выразить следующим образом:

$$|\xi_i - \xi_j| = d_{ij}^d. \quad (2)$$

Множества заданных расстояний между двумя агентами  $i$  и  $j$ ,  $d_{ij}^d$ , формируют заданную матрицу смежности  $A_d$ . Дополнительно ограничение на столкновение сформулировано в матрице  $A(t)$ :

$$\forall a_{ij}(t) \in A(t): a_{ij}(t) > c \text{ при } i \neq j, \quad (3)$$

где  $c \in \mathfrak{R}$  – минимально допустимое расстояние между любыми двумя соседними подсистемами. Чтобы проиллюстрировать это условие в виде целевой функции, задается новая функция  $\delta_{ij}(t)$  между каждым  $i$ -м БПЛА и его соседней подсистемой  $j$  [24]:

$$\delta_{ij}(t) = 1 + e^{-\frac{c - a_{ij}(t)}{\sigma}}. \quad (4)$$

Значение  $\delta_{ij}(t)$  зависит от разницы между расстоянием, разделяющим два  $i$  и  $j$  БПЛА, и безопасным расстоянием  $c$  (где  $l > c$ ) и сходится к единице, когда  $a_{ij}(t) \gg c$ . Значение  $\sigma$  постоянно между 0 и 1. Следует также предотвращать столкновения с препятствиями, что учитывается путем введения нового члена в целевую функцию:

$$\text{Obs}_{ik}(t) = e^{-\frac{c - d_{ik}}{\sigma}}. \quad (5)$$

Значение  $Obs_{ik}(t)$  является функцией разности между  $d_{ik}$ , расстоянием между  $i$ -м БПЛА и препятствием  $k$  и безопасным расстоянием  $c$ . Это значение стремится к нулю, когда расстояние значительно превышает безопасное.

### 2.3 Модель одного гексакоптера

В данном разделе выводятся уравнения движения гексакоптера (см. рис. 1), который представляет собой малоприводную систему с шестью степенями свободы и четырьмя входными параметрами. Аппарат состоит из шести актюаторов и имеет массу  $m$ . Для начала введем определения систем координат (СК). Пусть  $I = (X_p, Y_p, Z_p)$  – глобальная инерциальная СК, а  $B = (X_B, Y_B, Z_B)$  – корпусная СК. БПЛА подвержен воздействию семи сил: шести сил тяги  $f_i$ , действующих в направлении корпуса БПЛА  $Z_B = (0; 0; 1)$ , как показано на рис. 1, и силы тяжести  $mg$ . Дополнительно на БПЛА действуют семь крутящих моментов: по одному на каждый пропеллер  $\tau_i$  и момент силы сопротивления  $\tau_d$ , который в данном случае не учитывается, чтобы упростить модель.

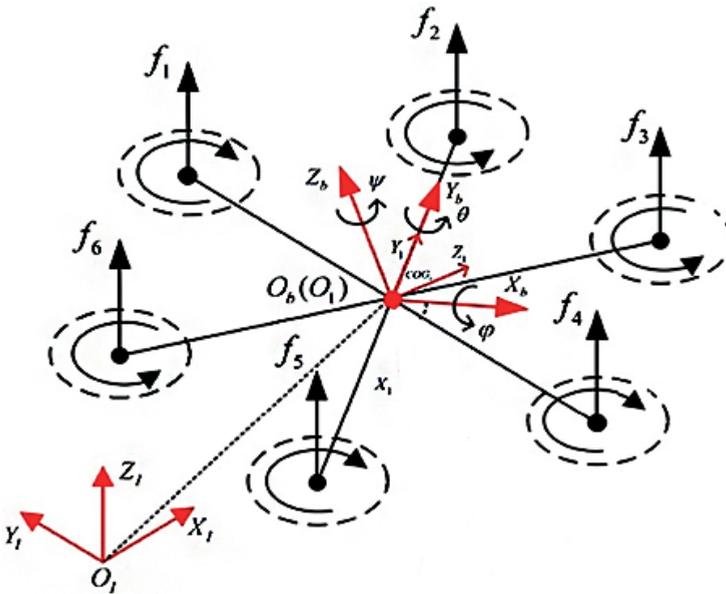


Рис. 1. Модель гексакоптера

Нелинейная модель гексакоптера описывается следующими уравнениями поступательного и вращательного движения:

$$\begin{aligned} \dot{q} &= v, \\ m\ddot{q} &= \mathcal{g} + RU, \\ \dot{\eta} &= W\Omega, \\ J\dot{\Omega} &= \Omega \times J\Omega + \tau_{res}. \end{aligned} \tag{6}$$

Векторы  $q = [x \ y \ z]^T$  и  $v = [v_x \ v_y \ v_z]^T$  – координаты и скорости центра масс, векторы  $\eta = [\phi \ \theta \ \psi]^T$  и  $\Omega = [W_{Bx} \ W_{By} \ W_{Bz}]^T$  – эйлеровы углы и угловые скорости,  $J$  – момент

инерции и  $\mathcal{g} = [0 \ 0 \ -g]^T$  – гравитационное ускорение.  $R$  – матрица поворота от корпусной к инерциальной СК,  $W$  – матрица преобразования скоростей.  $U = [0 \ 0 \ F]^T$  – вектор силы тяги,  $\tau_{res} = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T$  – вектор крутящего момента. Центр каждого винта находится на расстоянии  $L$  от центра масс, таким образом,

$$\tau_{res} = \begin{bmatrix} (f_3 + 2f_4 + f_5 - 2f_1 - f_2 - f_6) \cdot L / 2 \\ (f_3 + f_4 - f_1 - f_6) \cdot L / 2 \\ \tau_1 - \tau_2 - \tau_3 + \tau_4 + \tau_5 - \tau_6 \end{bmatrix}. \quad (7)$$

Сила тяги  $f_i$  и крутящий момент  $\tau_i$   $i$ -го пропеллера пропорциональны квадрату угловой скорости  $\omega_i$  с коэффициентами силы тяги  $k_f$  и крутящего момента  $k_t$ :

$$\begin{aligned} f_i &= k_f \omega_i^2, \\ \tau_i &= k_t \omega_i^2. \end{aligned} \quad (8)$$

### 2.4 Стратегия управления

Управление группой гексакоптеров разделено на два этапа. Первый заключается в управлении внутренней динамикой каждого гексакоптера в строе, под которой понимается динамика вращения и высоты  $z_i$ . В рассматриваемых сценариях задается фиксированный курс  $\psi_i$  и эталонная высота  $z_i$ . Выходные данные контроллеров  $x_i$  и  $y_i$ , полученные с использованием алгоритма планирования траектории на основе АУПР, подаются на входы контроллеров углов крена и тангажа. Архитектура управления показана на рис. 2.

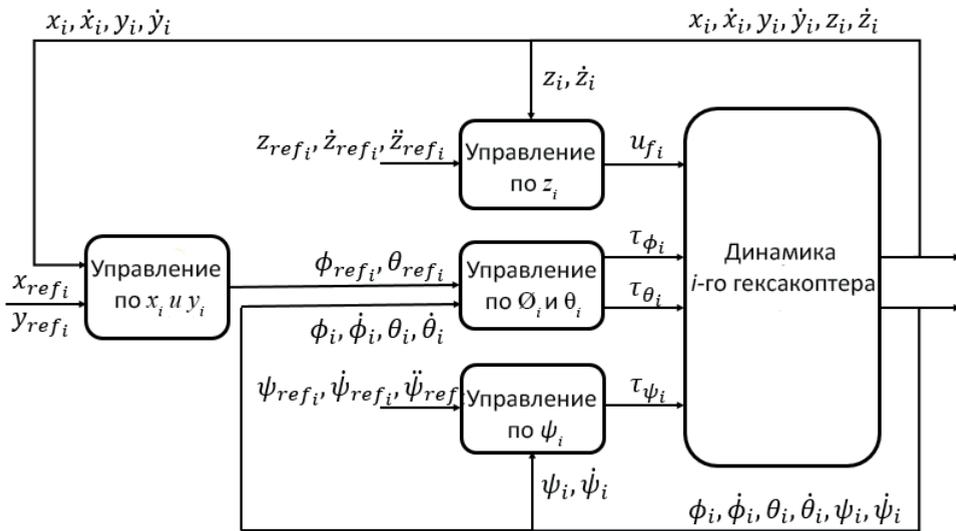


Рис. 2. Архитектура управления

Закон управления, применяемый к гексакоптеру, основан на функции насыщения для углов тангажа и крена, а также на ПИД-управлении по углу рыскания и высоте. Управление по функции насыщения гарантирует, что  $\phi, \dot{\phi}, \theta$  и  $\dot{\theta}$  остаются ограни-

ченными в течение всего полета [34]. Управляющие воздействия, применяемые для стабилизации динамики вращения  $\phi-\theta$ , описываются ниже:

$$\tau_{\phi} = \frac{J_x}{g} [\sigma_{py} (k_{py} (y - y_{ref})) + \sigma_{dy} (k_{dy} \dot{y}) - \sigma_{p\phi} (k_{p\phi} \phi) - \sigma_{d\phi} (k_{d\phi} \dot{\phi})], \quad (9)$$

$$\tau_{\theta} = \frac{J_y}{g} [\sigma_{px} (k_{px} (x - x_{ref})) + \sigma_{dx} (k_{dx} \dot{x}) - \sigma_{p\theta} (k_{p\theta} \theta) - \sigma_{d\theta} (k_{d\theta} \dot{\theta})], \quad (10)$$

где  $\sigma_{ai}(s)$  – функция насыщения, определяемая как

$$\begin{cases} \sigma_{\alpha_i}(s) = \alpha_i & \text{if } s > \alpha_i, \\ \sigma_{\alpha_i}(s) = s & \text{if } -\alpha_i < s < \alpha_i, \\ \sigma_{\alpha_i}(s) = -\alpha_i & \text{if } s < -\alpha_i; \end{cases} \quad (11)$$

$k_{py}$ ,  $k_{dy}$ ,  $k_{p\phi}$ ,  $k_{d\phi}$ ,  $k_{px}$ ,  $k_{dx}$ ,  $k_{p\theta}$  и  $k_{d\theta}$  – положительные коэффициенты усиления;  $x_{ref}$  и  $y_{ref}$  – заданные опорные значения, выведенные на шаге формирования траектории с использованием АУПР.

Для управления динамикой высоты и рыскания используются ПД-контроллеры. Управляющие воздействия данных двух подсистем задаются общим выражением для ПД-контроллера со слежением:

$$u_f = k_{pz} (z_{ref} - z) + k_{dz} (\dot{z}_{ref} - \dot{z}), \quad (12)$$

$$\tau_{\psi} = k_{p\psi} (\psi_{ref} - \psi) + k_{d\psi} (\dot{\psi}_{ref} - \dot{\psi}), \quad (13)$$

где  $\psi_{ref}$  и  $z_{ref}$  – заданные курс и высота гексакоптера, а коэффициенты  $k_{pz}$ ,  $k_{dz}$ ,  $k_{p\psi}$  и  $k_{d\psi}$  – положительные.

### 3. Алгоритм управления искусственным пчелиным роем

#### 3.1 Принцип АУПР

АУПР – это методика оптимизации, используемая для решения нелинейных задач поиска. В ее основе лежит поведение роя медоносных пчел при поиске источника пищи. Рой состоит из трех групп: рабочих пчел, пчел-наблюдателей и пчел-разведчиков. Позиция источника пищи представляет собой возможное решение задачи оптимизации, а количество нектара – пригодность этого решения. Количество решений эквивалентно количеству наблюдателей и рабочих пчел. Каждый цикл процесса поиска в АУПР состоит из четырех этапов [35].

**Этап инициализации.** АУПР вырабатывает беспорядочно распределенное исходное множество  $P$  из  $N_p$  решений (позиций источников пищи) и оценивает количество нектара (пригодность) каждого из них. Каждое решение  $\chi_i$  ( $i = 0, 1, 2, \dots, N_p - 1$ ) представляет собой  $D$ -размерный вектор, где  $D$  – количество параметров оптимизации. Инициализацию можно моделировать следующим образом:

$$\chi_{ij} = \chi_{ij}^{min} + rand(0,1) (\chi_{ij}^{max} - \chi_{ij}^{min}). \quad (14)$$

Пригодность источника пищи рассчитывается при помощи следующего выражения:

$$fit(\chi_i) = \frac{1}{1 + \Lambda(\chi_i)}, \quad (15)$$

где  $\Lambda(\chi_i)$  – значение целевой функции при  $\chi_i$ .

Размещение рабочих пчел у источников пищи. Рабочая пчела запоминает позицию в соответствии с локальной информацией и проверяет количество нектара у нового источника. Если нектара там больше, чем у старого источника, пчела переходит на новую позицию.

Размещение пчел-наблюдателей у источников пищи. После завершения задачи поиска рабочие пчелы передают сведения о решении (позицию источников пищи и количество нектара) пчелам-наблюдателям в зоне танца. Пчела-наблюдатель анализирует сведения о нектаре от всех рабочих пчел и выбирает источник пищи на основе значения его вероятности  $P_i$ , рассчитанного по следующему выражению:

$$P_i = \frac{fit(\chi_i)}{\sum_{i=1}^n fit(\chi_i)}. \quad (16)$$

После этого пчела-наблюдатель модифицирует позицию в памяти и исследует пригодность потенциального источника пищи. Если она выше, чем у предыдущего решения, пчела сохраняет новую позицию и удаляет старую. Чтобы найти решение  $\chi'_i$  вблизи  $\chi_i$ , случайным образом выбирается другое решение  $\chi_k$  и параметр решения  $j$ . Все переменные решения  $\chi'_i$  идентичны переменным решения  $\chi_i$ , кроме значения  $j$ -го параметра:

$$\chi'_i = (\chi_{i0}, \chi_{i1}, \chi_{i2}, \dots, \chi_{i(j-1)}, \chi'_{ij}, \chi_{i(j+1)}, \dots, \chi_{i(D-1)}). \quad (17)$$

Значение параметра  $\chi'_{ij}$  в решении  $\chi'_i$  вычисляется с помощью следующего выражения:

$$\chi'_{ij} = \chi_{ij} + u(\chi_{ij} - \chi_{kj}), \quad (18)$$

где  $u$  – равномерно распределенная случайная переменная в диапазоне  $[-1, 1]$ , а  $k$  – любое число от 0 до  $N_p - 1$ , отличное от  $i$ .

Этап разведки. Если не удастся улучшить позицию за заданное количество циклов (лимит), пчелы покидают источник пищи и переходят к новой позиции, случайным образом найденной пчелами-разведчиками.

Затем последние три этапа повторяются до тех пор, пока не будут выполнены критерии остановки.

### 3.2 Модифицированная версия АУПР

Согласно описаниям различных приложений, встречающимся в литературе, АУПР считается эффективным алгоритмом оптимизации. Однако ему присущ один недостаток, связанный с уравнением поиска решения, которое используется для выработки новых потенциальных решений на основе данных о предыдущих. В соот-

ветствии с уравнением поиска решения (18) новое потенциальное решение находится путем смещения от старого к другому решению, случайным образом выбранному из множества. Новое решение может быть с равной степенью вероятности как хорошим, так и плохим, т.е. нельзя с уверенностью сказать, что оно будет лучше предыдущего. С другой стороны, коэффициент  $u$  в уравнении (18) является равномерно распределенным случайным числом в диапазоне  $[-1, 1]$ , а  $\chi_{kj}$  – случайный элемент множества. Таким образом, поиск решения, задаваемый уравнением (18), является случайным для исследования [36]. Чтобы повысить качество исследования, в уравнении поиска вводятся данные о глобально лучшем решении для выбора направления поиска потенциальных решений, тогда его можно записать следующим образом:

$$\chi'_{ij} = \chi_{ij} + u(\chi_{ij} - \chi_{kj}) + \Psi(y_j - \chi_{ij}). \quad (19)$$

Здесь в правой части добавлен третий член «глобального улучшения»,  $y_j$  –  $j$ -й элемент глобально лучшего решения,  $\Psi$  – равномерно распределенное случайное число в диапазоне  $[0, C]$ , где  $C$  – неотрицательная заданная постоянная, равная 1,5 в соответствии с [36]. Член «глобального улучшения» может новый вариант решения довести до глобально лучшего решения, что позволит повысить качество работы АУПР. Ниже приведен псевдокод АУПР-ГЛ.

---

**Алгоритм 1.** Оптимизация целевой функции согласно АУПР-ГЛ

Инициализация алгоритма

**WHILE** (Iter < Max Iterations)

**FOR** ( $i = 1: N_p$ )

    Нахождение нового источника пищи

    Расчет пригодности нового источника пищи

    Жадный выбор

**END FOR**

    Расчет вероятности для каждого источника пищи  $P_i$

**FOR** ( $i = 1: N_p$ )

    Пчелы-наблюдатели находят новые источники пищи в зависимости от  $P_i$

    Нахождение нового источника пищи

    Оценка пригодности нового источника пищи

    Жадный выбор (принятие локально оптимального решения на каждом этапе, допуская, что конечное решение также окажется оптимальным)

**END FOR**

**IF** (любая рабочая пчела превращается в разведчика)

    Отсылка пчелы-разведчика к случайным образом найденному источнику пищи

**END IF**

Запись лучшего решения

Iter = Iter + 1;

**END WHILE**

---

### 3.3 Применение АУПР к задаче группового полета БПЛА

Ниже приводится постановка задачи формирования строя гексакоптеров, где целевая функция для каждого аппарата определяется, как и в работе [24], следующим выражением:

$$\Lambda_i(t) = \rho \left( \|P_d - [\xi_i(t) + h]\| - d_{ip}^d \right) + \sum_{j=0}^q k_j(t) \delta_{ij}(t) \left( \|\xi_i(t) - [\xi_i(t) + h]\| - d_{ij}^d \right) + \sum_{k=1}^n Obs_{ik}(t), \quad (20)$$

где  $i \neq j$ ,  $q = \text{card}(\epsilon_i)$ ,  $h \in \mathbb{R}^2$  и  $\rho \gg 1$ ,  $d_{ip}^d$  – заданное расстояние между  $i$ -м БПЛА и точкой встречи  $P_d$ ,  $n$  – количество обнаруженных препятствий. Задача состоит в том, чтобы получить наилучший вектор  $h$  для каждого  $i$ -го БПЛА, который минимизирует целевую функцию  $\Lambda_i(t)$ , так что  $\forall i \in N: \lim_{t \rightarrow \infty} \Lambda_i(t) = 0 \Rightarrow \lim_{t \rightarrow \infty} A(t) = A_d$ . Заданная траектория в момент  $t+t_d$  для  $i$ -го БПЛА описывается выражением

$$Y_{i \text{ ref}}(t+t_d) = Y_{i \text{ ref}}(t) + h, \quad (21)$$

где  $t_d$  – временной шаг. Коэффициент  $\rho$  должен быть больше 1. Такой выбор обусловлен необходимостью ориентировать каждый  $i$ -й БПЛА в направлении  $P_d$ . Применение АУПР-ГЛ в задаче группового полета БПЛА представлено ниже.

---

**Алгоритм 2.** Выработка траектории с помощью АУПР-ГЛ

Инициализация параметров

Определение матрицы  $A_d$

**РЕПЕАТ** (на каждом временном этапе):

- оценка матрицы  $A(t)$ ;
- нахождение всех топологических соседей для  $i$ -го гексакоптера по набору  $\epsilon_i$ ;
- выбор наилучшего вектора  $[h_x \ h_y]$ , который оптимизирует целевую функцию  $\Lambda_i(t)$  при помощи АУПР-ГЛ;
- вычисление следующей позиции  $(x_{id}, y_{id})$  в момент  $t + t_d$ :

$$x_{id}(t+t_d) = x_i(t) + h_x,$$

$$y_{id}(t+t_d) = y_i(t) + h_y.$$

**UNTIL** Конец моделирования

---

## 4. Отказоустойчивость

Описывается применение предложенного в разделе 3 метода управления при возникновении сбоев. Особый интерес представляют случаи потери БПЛА вследствие отказов актюаторов в строе как с последующим восстановлением неисправных аппаратов, так и без него.

#### 4.1 Применение метода управления при возникновении сбоев без восстановления неисправного аппарата

Ниже предлагается реализация модуля реконфигурации целевых функций и заданной матрицы смежности, присутствующих в модулях оптимизации, для отработки отказа, вызванного потерей аппарата в строе. В модуле реконфигурации (рис. 3) использованы работоспособные модули диагностики отказов, позволяющие воздействовать на целевые функции, чтобы модифицировать траектории, рассчитанные с помощью модулей оптимизации.

Ниже рассматривается ситуация потери аппарата в строе из  $N$  БПЛА. Рассмотрим диагональную матрицу  $K(t)$  из  $N \times N$  элементов  $k_i(t)$ , где

$k_i(t) = 1$ , если  $i$ -й аппарат исправен,  
 $k_i(t) = 0$ , если  $i$ -й аппарат неисправен.

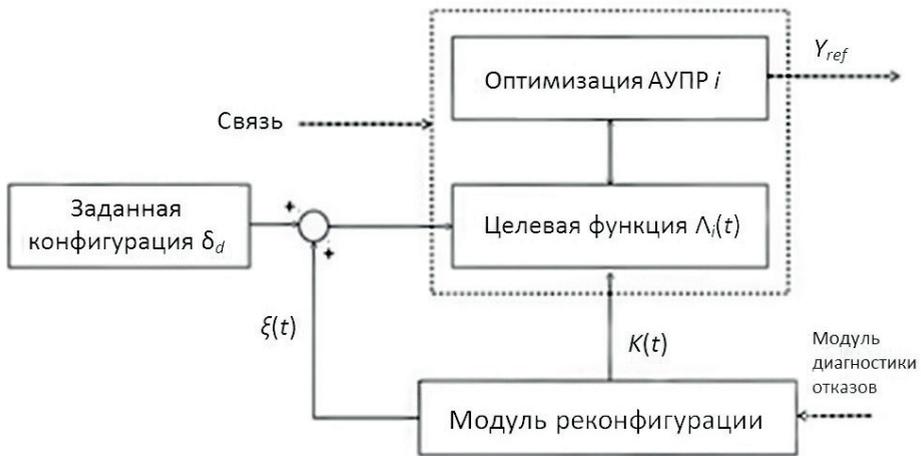


Рис. 3. Схема реконфигурации модулей оптимизации

После модификации (20) целевая функция принимает следующий вид:

$$\Lambda_i(t) = \rho \left( \left| P_d - [\xi_i(t) + h] \right| - d_{ip}^d \right) + \sum_{j=0}^q k_j(t) \delta_{ij}(t) \left( \left| \xi_i(t) - [\xi_i(t) + h] \right| - d_{ij}^d \right) + \sum_{k=1}^n Obs_{ik}(t). \quad (22)$$

При нулевом значении параметра  $k_j(t)$  исключается влияние  $i$ -го аппарата на выработку траектории остальных аппаратов строя. Тогда потеря БПЛА будет учитываться в целевых функциях модулей оптимизации, что значительно облегчает управление строем.

#### 4.2 Восстановление неисправного аппарата путем распределения управления

В данном разделе рассматривается восстановление неисправного аппарата в строе при обнаружении отказа. Остальные БПЛА должны продолжать выполнять задание,

чтобы достичь заданной топологии, как указано в разделе 3, без реконфигурирования модуля оптимизации, таким образом, целевая функция и заданная матрица смежности остаются неизменными. Для восстановления неисправного гексакоптера предлагается использовать распределение отказоустойчивого управления.

Важное преимущество гексакоптера заключается в наличии избыточного количества двигателей. Вследствие этого неисправный аппарат может продолжать полет даже при отказе одного из двигателей. Подробный анализ управляемости гексакоптеров дан в работе [37]. Распределение управления позволяет добиться желаемого эффекта управления за счет избыточного количества актюаторов. В частности, если один или несколько актюаторов отказали, можно адаптировать схему распределения управления таким образом, чтобы предотвратить участие неисправных приборов в выработке управляющего воздействия. При этом ключевое значение для успешного распределения управления имеет адекватная информация о произошедших сбоях. В данном случае полагается, что работоспособный диагностический модуль выявляет отказ и корректно его локализует. Под отказом понимается неисправность актюатора в одном из шести двигателей гексакоптера.

Как и в том случае, когда неисправный аппарат не восстанавливается, диагностический модуль вырабатывает сигнал об обнаружении отказа. Этот сигнал поступает на схему распределения отказоустойчивого управления, которая воздействует на управляющую командную систему БПЛА и тем самым исключает влияние неисправного элемента в диагональной весовой матрице  $Y(t)$  размером  $6 \times 6$  элементов  $\gamma_i(t)$ :

- $\gamma_i(t) = 1$ , если двигатель  $i$ -го аппарата исправен,
- $\gamma_i(t) = 0$ , если двигатель  $i$ -го аппарата неисправен.

Для решения линейных задач распределения управления без ограничений обычно применяется метод псевдообратного. Игнорируя любые ограничения и насыщения входного воздействия, задачу распределения управления можно решить путем оптимизации квадратичной задачи

$$\min_v \left| Y^{-1} v \right| \text{ s.t. } Bv(t) = \tau \quad (23)$$

и получить решение на основе взвешенного псевдообратного

$$v = \left[ YB^T (BYB^T)^{-1} \right] \cdot \tau, \quad (24)$$

где  $\tau = [u_f \tau_\phi \tau_\theta \tau_\psi]^T$  – вектор виртуальных управляющих воздействий,  $v = [\omega_1^2, \dots, \omega_6^2]^T$  – вектор квадратов скоростей двигателей, а  $B$  – матрица эффективности управления, связывающая виртуальные управляющие воздействия со скоростями двигателей. Фиксированная матрица  $B$  определяется как

$$B = \begin{bmatrix} t_1 & \dots & t_6 \\ r_1 & \dots & r_6 \\ p_1 & \dots & p_6 \\ y_1 & \dots & y_6 \end{bmatrix}, \quad (25)$$

где

$$\begin{aligned}
 t_i &= k_f \\
 r_i &= \begin{cases} (-1)^i k_f L & i = 1, 4 \\ (-1)^{i+1} k_f L / 2 & i = 2, 3, 5, 6 \end{cases} \\
 p_i &= \begin{cases} \frac{\sqrt{3}}{2} k_f L & i = 3, 4 \\ -\frac{\sqrt{3}}{2} k_f L & i = 1, 6 \\ 0 & i = 2, 5 \end{cases} \\
 y_i &= \begin{cases} k_t & i = 1, 4, 5 \\ -k_t & i = 2, 3, 6 \end{cases}
 \end{aligned} \tag{26}$$

### 5. Результаты моделирования

Испытания стратегии управления АУПР-ГЛ проведены путем моделирования на симуляторе строя гексакоптеров в среде Matlab. Используемые параметры алгоритма приведены в табл. 1.

Т а б л и ц а 1

Параметры АУПР-ГЛ

Параметр	Значение
Размер множества	100
Лимит	100
Максимальное количество итераций	1000
$H_{max}$	-0,25
$H_{min}$	0,25
Допуск	0,1
Безопасное расстояние	1

Вначале для выработки траекторий 6 БПЛА при известных начальных позициях и заданной точке (10, 15) при сохранении круговой компоновки радиуса 6 м<sup>1</sup> был применен метод АУПР. В строе может присутствовать любое количество аппаратов более одного. Число БПЛА влияет только на матрицу смежности  $A_d$ , и алгоритм работает так, как описано в данной статье. Полученные траектории показаны на рис. 4, а. На рис. 4, б представлено схождение целевой функции каждого аппарата, которое занимает около 80 с. Тот же сценарий испытывался на модифицированном АУПР-ГЛ, сформированные траектории и схождение целевой функции продемонстрированы на рис. 4, в и г соответственно. Показано, что АУПР-ГЛ работает быстрее, чем АУПР: время сходимости сокращается до 60 с.

<sup>1</sup> Видео моделирования доступно в интернете: <https://youtu.be/sMTEfGGa58k>.

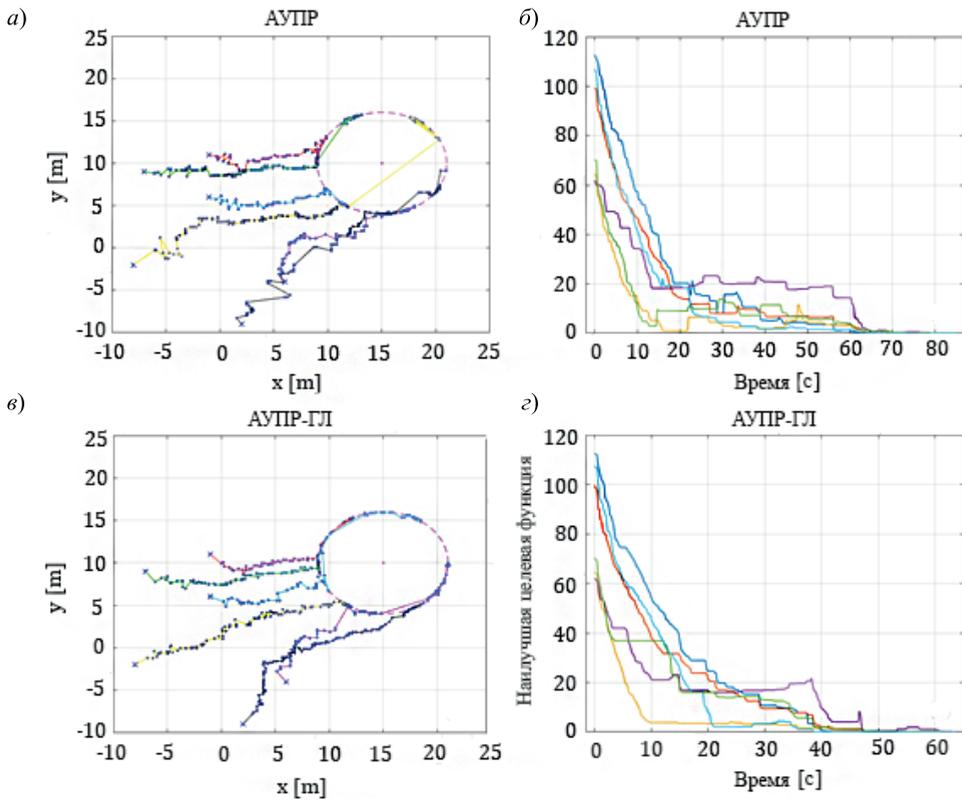


Рис. 4: *a* – траектории шести БПЛА при облете препятствия (АУПР),  
*б* – значения целевых функций для шести БПЛА при облете препятствия (АУПР),  
*в* – траектории шести БПЛА при облете препятствия (АУПР-ГЛ),  
*г* – значения целевых функций для шести БПЛА при облете препятствия (АУПР-ГЛ)

Испытания по описанному выше сценарию проводились на симуляторе гексакоптера в программе Simulink с учетом нелинейной динамики модели и наличия препятствий на траекториях. Наличие препятствий моделируется произвольной известной позицией, которую БПЛА должны облететь. В реальных полетах препятствия можно обнаружить при помощи различных сенсоров: камер, лидаров и ультразвуковых датчиков. Для управления положением и высотой применяется ПИД-контроллер. На рис. 5, *a* и *б* показаны заданные расстояния между БПЛА в каждой паре, рассчитанные с помощью АУПР-ГЛ, и реальные расстояния, рассчитанные с использованием симулятора. Ограничение по уклонению от препятствий выполняется, так как все эти расстояния превышают безопасное расстояние  $c$ . На рис. 5, *в* и *г* показано, что ограничение по уклонению от препятствий выполняется также за счет соблюдения дистанции между каждым аппаратом и препятствием в соответствии с расчетными позициями.

Качество работы АУПР-ГЛ сравнивается с качеством работы алгоритма ОРЧ, описанного, например, в [24]. Оба алгоритма испытывались по четырем различным сценариям с одинаковыми условиями. Было проведено 100 моделирований и рассчитано среднее время сходимости для каждого алгоритма при разных сценариях. Результаты приведены в табл. 2. Из них следует, что сходимость АУПР-ГЛ происходит быстрее, чем ОРЧ, а также что АУПР-ГЛ эффективен, обладает меньшим временем сходимости и подходит для рассматриваемых применений.

Среднее время сходимости АУПР-ГЛ и ОРЧ

Сценарий	АУПР-ГЛ	ОРЧ
Один аппарат без препятствия	5,4 с	6,3 с
Один аппарат с препятствием	5,06 с	6,84 с
Два аппарата	11,4 с	12,38 с
Шесть аппаратов	108 с	160 с

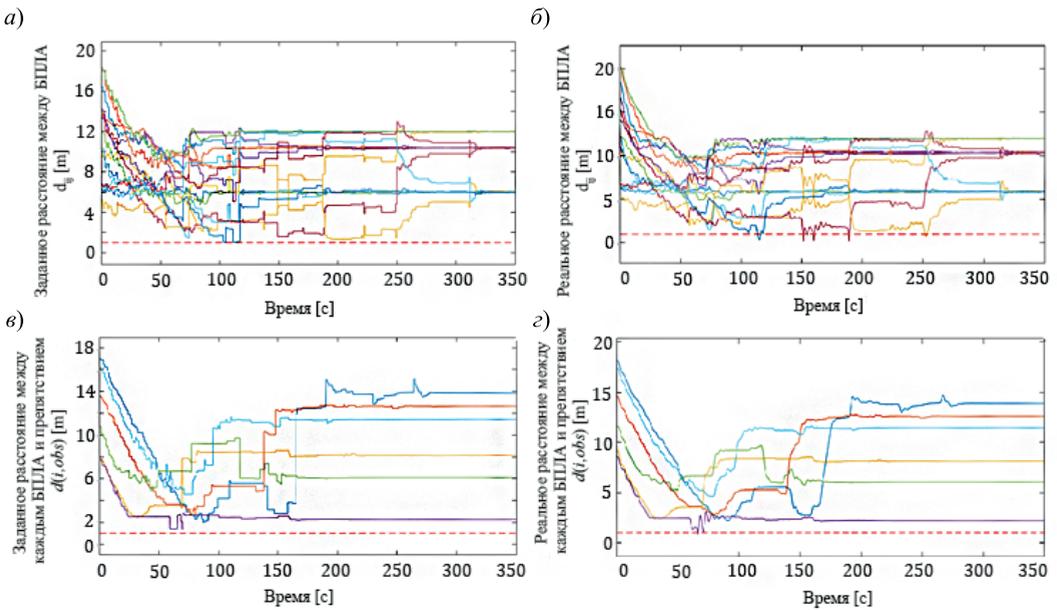


Рис. 5: а – расстояние между БПЛА в каждой паре (АУПР-ГЛ),  
 б – реальное расстояние между БПЛА в каждой паре,  
 в – расстояние между каждым БПЛА и препятствием (АУПР-ГЛ),  
 з – реальное расстояние между каждым БПЛА и препятствием

Для испытаний предлагаемых методик отказоустойчивого управления было использовано несколько сценариев. Если неисправный аппарат не был восстановлен, предполагалось, что в момент  $t = 50$  с обнаруживается отказ БПЛА 4. Он совершает вертикальную посадку и далее остается неподвижным. Оставшиеся пять БПЛА продолжают свой полет с учетом того, что модуль реконфигурации обновляет модуль оптимизации таким образом, чтобы пять исправных аппаратов скомпоновались вокруг целевой точки без столкновений. На рис. 6 видно, что отказ БПЛА 4 обнаруживается после 50 с моделирования, и БПЛА остается в том же положении. Оставшиеся пять гексакоптеров продолжают выполнение задания и двигаются строем в направлении новых расчетных позиций. Сбой показан вертикальными пунктирными линиями. На рис. 7 приведены траектории перегруппировки от исходных до финальных позиций для каждого аппарата, рассчитанные с помощью АУПР-ГЛ. Неисправный БПЛА перестает двигаться.

В случае восстановления неисправного аппарата предполагался отказ второго двигателя БПЛА 4 в строю на 40-й секунде после начала моделирования. Поведение строя тестировалось на Matlab-симуляторе после восстановления неисправного БПЛА при распределении отказоустойчивого управления. За счет распределе-

ния управления удалось исключить влияние неисправного винта, модифицировать управляющее воздействие на БПЛА 4 и восстановить его в течение одной секунды. Отслеживание положения БПЛА 4 ( $x$  и  $y$ ) показано на рис. 8. После восстановления БПЛА 4 отслеживает заданную траекторию и продолжает полет с другими членами строя, чтобы достичь намеченной цели. Следует отметить, что предлагаемое распределение управления позволяет отработать до двух отказов двигателей в конфигурации с вращающимися в одном направлении винтами [38].

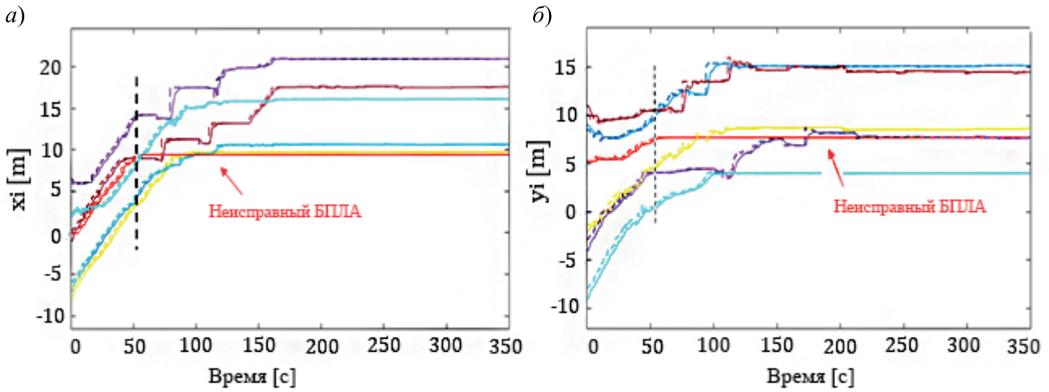


Рис. 6. Заданные и реальные позиции шести БПЛА при обнаружении неисправного аппарата в полете:  $a - X$ ,  $b - Y$

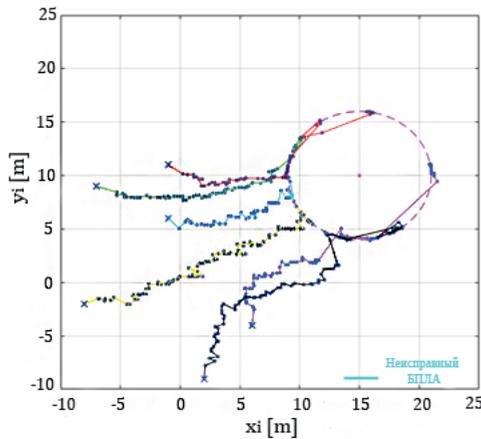


Рис. 7. Траектории шести БПЛА при обнаружении неисправного аппарата в полете, выработанные алгоритмом АУПР-ГЛ

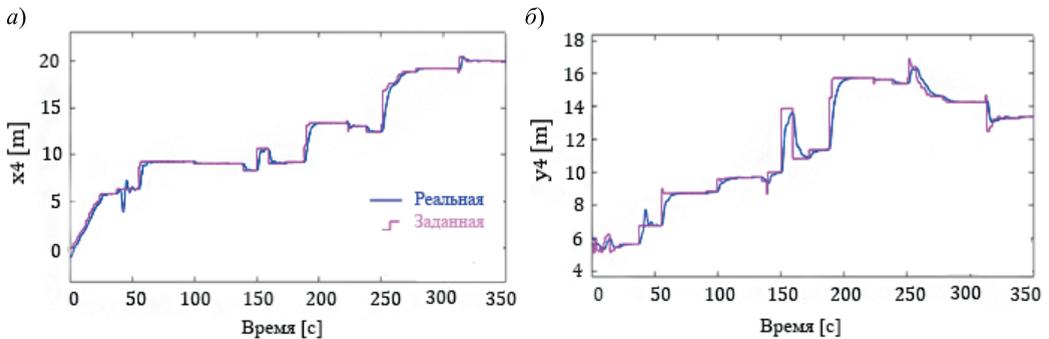


Рис. 8. Отслеживание позиции неисправного БПЛА:  $a - X$ ,  $b - Y$

## 6. Заключение

В работе представлен алгоритм распределенного планирования траекторий для формирования и управления групповыми полетами БПЛА в режиме реального времени. В его основе лежит АУПР-ГЛ, решающий задачу оптимизации для каждого БПЛА, чтобы минимизировать целевую функцию и рассчитать оптимальную опорную траекторию. Это обеспечивает такое управление строем БПЛА, при котором аппараты достигают заданной цели в нужной топологии, избегая столкновений и совершая облет препятствий. Целевая функция для каждого БПЛА минимизируется по распределенному принципу.

Результаты моделирования гексакоптеров в Matlab продемонстрировали, что предлагаемый метод обеспечивает отслеживание цели и предотвращает столкновения. Комбинирование локальных законов управления, отказоустойчивых алгоритмов и распределенного планирования траекторий позволяет добиться высокой робастности к отказам актюаторов и потере аппаратов.

В дальнейшем планируется экспериментально испытать предлагаемую архитектуру на строе реальных гексакоптеров внутри помещения, а также ее эффективность при работе с другими типами отказов, такими как неисправности телекоммуникаций, потери данных, задержки, отказы датчиков и погрешности определения местоположения.

## Благодарности

*Работа выполнена в рамках Программы поддержки исследований  
Ливанского университета.*

## ЛИТЕРАТУРА

1. **Shakhtrah, H., Sawalmeh, A., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N., Khreishah, A., and Guizani, M.,** Unmanned Aerial Vehicles (UAVs): A survey on civil applications and key research challenges, *IEEE Access* 7, 2019, pp. 48572-48634.
2. **Sreenath, K. and Kumar, V.,** Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots, in *Robotics: Science and Systems*, Berlin, 2013.
3. **Franchi, A., Secchi, C., Ryll, M., Bulthoff, H., and Giordano, P.,** Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs, *IEEE Robot. Autom. Mag.*, 2012, vol. 19, no. 3, pp. 57–68.
4. **Ritz, R. and D’Andrea, R.,** Carrying a flexible payload with multiple flying vehicles, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 3-7 November 2013, pp. 3465–3471.
5. **Jiang, Q. and Kumar, V.,** The inverse kinematics of cooperative transport with multiple aerial robots, *IEEE Trans. Robot.*, 2013, vol. 29, no. 1, pp. 136–145.
6. **Kushleyev, A., Kumar, V., and Mellinger, D.,** Towards a swarm of agile micro quadrotors, *Proc. Robot., Sci. Syst.*, Sydney, NSW, Australia, 2012.
7. **Michael, N., Fink, J., and Kumar, V.,** Cooperative manipulation and transportation with aerial robots, *Proc. Robot., Sci. Syst.*, Seattle, WA, USA, 2009.
8. **Dames, P. and Kumar, V.,** Autonomous localization of an unknown number of targets without data association using teams of mobile sensors, *IEEE Trans. Autom. Sci. Eng.*, 2015, vol. 12, no. 3, pp. 850–864.
9. **Saska, M. et al.,** Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance, *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Orlando, FL, USA, 27-30 May 2014, pp. 584–595.
10. **Hou, Z. and Fantoni, I.,** Interactive leader-follower consensus of multiple quadrotors based on composite nonlinear feedback control, *IEEE Transactions on Control Systems Technology*, 2018, vol. 26, no. 5, pp. 1732–1743.

11. Vasarhelyi, G., Viragh, Cs., Somorjai, G., Tarcai, N., Szorenyi, T., Nepusz, T., and Viscek, T., Outdoor flocking and formation flight with autonomous aerial robots, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, 14–18 September 2014.
12. Scollig, A., Augugliaro, F., Lupashin, S., and D'Andrea, R., Synchronizing the motion of a quadcopter to music, *IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, May 3–8 2010, pp. 3355–3360.
13. Kushleyev, A., Mellinger, D., Powers, C., and Kumar, V., Towards a swarm of agile micro quadrotors, *Autonomous Robots*, 2013, vol. 35, no. 4, pp. 287–300.
14. Reynolds, C., Flocks, Herds, and Schools: A distributed behavioral model, *Proc. of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, 1987, pp. 25–34.
15. Olfati-Saber, R., Flocking for multi-agent dynamic systems: algorithms and theory, *IEEE Transactions on Automatic Control*, 2007, vol. 51, pp. 863–868.
16. Antonelli, G., Arrichiello, F., and Chiaverini, S., Flocking for multirobot systems via the nullspace based behavioral control, *Swarm Intelligence*, 2010, vol. 4, no. 37.
17. Bellingham, J., Tillerson, M., Alighanbari, M., and How, J., Cooperative path planning for multiple UAVs in dynamic and uncertain environments, *Proc. IEEE Conference on Decision and Control*, Las Vegas, NV, USA, 2002.
18. Richards, A. and How, J., Aircraft trajectory planning with collision avoidance using mixed integer linear programming, *Proc. IEEE American Control Conference*, Anchorage, AK, USA, 2002.
19. Tanner, H., Jadbabaie, A., and Pappas, G., Flocking in fixed and switching networks, *IEEE Transactions on Automatic Control*, 2007, vol. 52, no. 5, pp. 863–868.
20. Bakule, L., Decentralized control: An overview, *Annual Reviews in Control*, 2008, vol. 32, no. 1, pp. 87–98.
21. Jovanovic, M., Modeling, analysis, and control of spatially distributed systems, PhD Thesis, University of California, Santa Barbara, 2004.
22. Brunet, L., Consensus-based auction approaches for decentralized task assignment, *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, 2008.
23. Quin, J., and Yu, C., Cluster consensus control of generic linear multi-agent systems under directed topology with acyclic partition, *Automatica*, 2013, vol. 49, no. 9, pp. 2898–2905.
24. Belkadi, A., Ciarletta, L., and Theilliol, D., UAVs fleet control design using distributed particle swarm optimization: A leaderless approach, *International Conference on Unmanned Aircraft Systems (ICUAS)*, Arlington, VA, USA, 2016.
25. Belkadi, A., Abaunza, H., Ciarletta, L., Castillo, P., and Theilliol, D., Distributed path planning for controlling a fleet of UAVs: Application to a team of quadrotors, *IFAC-PapersOnline*, 2017, vol. 50, no. 1, pp. 15983–15989.
26. Abdmouleh, Z., Gastli, A., Ben-Brahim, L., Haouari, M., and Al-Emadi, N., Review of optimization techniques applied for the integration of distributed generation from renewable energy sources, *Renewable Energy*, 2017, vol. 113, pp. 266–280.
27. Karaboga, D. and Basturk, B., Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems, *Foundations of Fuzzy Logic and Soft Computing*, Berlin: Springer-Verlag, 2007, pp. 789–798.
28. Bhattacharjee, P., Rakshit, P., Goswami, I., Konar, A., and Nagar, A., Multi-robot path-planning using artificial bee colony optimization algorithm, *Proc. World Congress on Nature and Biologically Inspired Computing*, Salamanca, Spain, 2011.
29. Soyinka, O. and Duan, H., Satellite formation keeping via chaotic artificial bee colony, *Aircr. Eng. Aerosp. Technol.*, 2017, vol. 89, no. 2, pp. 246–256.
30. Zhou, B., Wang, W., and Ye, H., Cooperative control for consensus of multi-agent systems with actuator faults, *Computers & Electrical Engineering*, 2014, vol. 40, no. 7, pp. 2154–2166.
31. Saska, M., Krajcnik, T., Vonasek, V., Kasl, Z., Spurny, V., and Preucil, L., Fault-tolerant formation driving mechanism designed for heterogeneous MAVs-UGVs groups, *Journal of Intelligent & Robotic Systems*, 2014, vol. 73, no. 1, pp. 603–622.
32. Belkadi, A., Conception de commande tolerante aux defauts pour les systemes multi-agents: application au vol en formation d'une flotte de vehicules autonomes aeriens, Ph.D. dissertation, University of Lorraine, France, 2017.
33. Diestel, R., *Graph Theory. Graduate Texts in Mathematics*, Heidelberg: SpringerVerlag, 2005, third edition.

34. Sanahuja, G., Castillo, P., and Sanchez, A., Stabilization of n integrators in cascade with bounded input with experimental application to a VTOL laboratory system, *International Journal of Robust and Nonlinear Control*, 2010, vol. 20, no. 10, pp. 1129–1139.
35. Lalitha, M., Reddy, N., and Reddy, V., Optimal DG placement for maximum loss reduction in radial distribution system using ABC algorithm, *Int. J. Rev. Comput.*, 2010, vol. 3, pp. 44–52.
36. Zhu, G. and Knowg, S., Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics and Computation*, 2010, vol. 217, no. 7, pp. 3166–3173.
37. Saied, M., Knaiber, M., Mazeh, H., Shraim, H., and Francis, C., BFA fuzzy logic based control allocation for fault-tolerant control of multirotor UAVs, *The Aeronautical Journal*, 2019, vol. 123, pp. 1356–1373.
38. Mazeh, H., Saied, M., Shraim, H., and Francis, C., Fault-tolerant control of an hexarotor unmanned aerial vehicle applying outdoor tests and experiments. *IFAC-PapersOnline*, 2018, vol. 51, no. 22, pp. 312–317.

---

Slim, M., Saied, M., Mazeh, H., Shraim, H., and Francis, C. (Scientific Research Center in Engineering, Faculty of Engineering, Lebanese University, Hadath, Lebanon)  
Fault-Tolerant Control Design for Multirotor UAVs Formation Flight, *Giroskopiya i Navigatsiya*, 2021, vol. 29, no. 2 (113), pp. 78–96.

**Abstract.** This paper proposes the use of the Gbest-Guided Artificial Bee Colony (GABC) algorithm to solve an online optimization problem for the leaderless distributed formation control of hexarotor Unmanned Aerial Vehicles (UAVs). The GABC is employed to optimize a cost function for each agent while ensuring the convergence of the fleet to the target position and averting both obstacles and collisions with other UAVs. The GABC algorithm has been shown to be competitive with some other conventional biological-inspired algorithms such as the Particle Swarm Optimization (PSO). Fault-Tolerant Control (FTC) methods are presented and tested on several scenarios, particularly we considered the cases of loss of agents and actuator faults in the fleet. Results show the success of the proposed FTC methods to minimize the faults effect on the formation final goal.

**Key words:** unmanned aerial vehicles, Artificial Bee Colony algorithm, formation control, fault-tolerant control.

Материал поступил 05.04.2021