

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук

Кафедра цифровых технологий

*Алгоритмы определения пространственной ориентации подвижных объектов в
задачах*

ВКР Магистерская диссертация

02.03.01 Математика и компьютерные науки

Распределенные системы и искусственный интеллект

Допущено к защите в ГЭК

Зав. кафедрой _____ С.Д. Кургалин, д. ф.-м. н., профессор _____.2017

Обучающийся _____ Е.Ю. Морозов, 2 курс, д/о

Руководитель _____ Л.А. Минин, к. ф.-м. н., доцент

Воронеж 2019

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных наук
Кафедра цифровых технологий

УТВЕРЖДАЮ
заведующий кафедрой

_____ Кургалин С. Д.,
подпись расшифровка подписи

ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
ОБУЧАЮЩЕГОСЯ МОРОЗОВА ЕВГЕНИЯ ЮРЬЕВИЧА

фамилия, имя, отчество

1. Тема работы «Алгоритмы определения пространственной ориентации подвижных объектов в задачах»,
утверждена решением ученого совета факультета компьютерных наук от ____ . ____ .20 ____
2. Направление подготовки 02.03.01 Математика и компьютерные науки.
шифр, наименование
3. Срок сдачи студентом законченной работы ____ . ____ .20 ____
4. Календарный план:

№	Структура ВКР	Сроки выполнения	Примечание
1	Введение	03.09.2018	
2	Глава 1.	10.09.2018	
3	Глава 2.	12.10.2018	
4	Глава 3.	02.02.2019	
5	Глава 4.	15.02.2019	
6	Заключение	13.05.2019	
7	Список использованных источников	03.06.2019	

Обучающийся

_____ Морозов Е. Ю.
подпись расшифровка подписи

Руководитель

_____ Минин Л. А.
подпись расшифровка подписи

Реферат

Магистерская диссертация 31 с., 0 источников, 2 приложения.

АЗИМУТАЛЬНО-УГЛОМЕСТНОЕ РАДИОПЕЛЕНГОВАНИЕ, ПРОСТРАНСТВЕННАЯ ОРИЕНТАЦИЯ, ДЕКАРТОВА СИСТЕМА КООРДИНАТ, КУРС, ТАНГАЖ, КРЕН, МАТРИЦА ВРАЩЕНИЯ.

Объект исследование — TODO.

Цель работы — TODO.

Метод исследования и аппаратура — TODO, теория алгоритмов, математический пакет символьных вычислений Wolfram Mathematica, персональный компьютер.

TODO.

Содержание

Введение	4
1 Обзор существующих алгоритмов	6
2 Постановка задачи	7
3 Локальные навигационные системы	9
3.1 Локальная угломерная навигационная система первого типа	9
3.2 Случай локальной системы с НПУ	11
3.3 Локальная система с уменьшенным количеством наземных РО	14
3.4 Случай автономной системы	17
4 Обсуждение результатов	22
Заключение	23
Список использованных источников	24
Приложение А Определение систем координат согласно ГОСТ	24
Приложение Б Листинг программы	25

Введение

TODO

1 Обзор существующих алгоритмов

TODO

2 Постановка задачи

Пусть N радиоориентиров размещены в i -х точках M_i пространства (где $i = 1, \dots, N$) с известными координатами $M_i(x_i, y_i, z_i)$ в нормальной земной системе координат (НЗСК). Подвижный объект находится в точке M_0 с неизвестными координатами $M_0(x, y, z)$, а его угловая ориентация определяется тремя неизвестными углами Эйлера: углами курса ψ , крена μ и тангажа ϑ . В результате азимутально-угломестного радиопеленгования реперного источника M_i , определяются углы азимута (α_i) и угла места (ε_i) в связанной системе координат подвижного объекта. Необходимо определить минимальное количество радиоориентиров M_i , при котором представляется возможным однозначно найти координаты $M_0(x, y, z)$ и угловую ориентацию (углы ψ , μ и ϑ) подвижного объекта.

Так как положение и угловая ориентация бортовой пеленгаторной антенны (БПА) определяется шестью параметрами (три координаты x, y, z и три угла ψ, μ и ϑ), то для их определения необходимо иметь как минимум шесть измеряемых величин. Этими величинами являются пары азимутов α_i и углов места ε_i . Таким образом, для однозначного определения положения и пространственной ориентации подвижного объекта, число радиоориентиров должно быть не менее трех.

Детерминированный подход к определению координат и пространственной ориентации подвижного объекта можно представить в виде трех-этапной процедуры, заключающейся в следующем:

1. Нахождение совокупности расстояний от фазового центра (ФЦ) БПА до радиоориентиров;
2. Определение координат подвижного объекта;
3. Нахождение матрицы вращения и связанных с ней углов Эйлера, определяющих угловую ориентацию БПА.

Следует отметить, что задачи второго и третьего этапов являются стандартными для радионавигации подвижных объектов, поэтому для их решения не требуется разрабатывать специальных методов. Ключевым оказывается именно первый этап.

Система уравнений для нахождения расстояний от ФЦ БПА до радиоориентиров на первом этапе нелинейна. При произвольном взаимном распо-

ложении БПА и радиоориентиров, решение упомянутой системы не однозначно, и может включать в себя от одного до четырех наборов расстояний, что ведет к неоднозначности определений координат и угловой ориентации. По мере увеличения числа источников, происходит переопределение системы, и, соответственно, она перестает быть совместной при малейших погрешностях измерений азимутов и углов места. Использование стандартного метода наименьших квадратов приводит к повышению степени уравнений в системе и, как следствие, к резкому усложнению процедуры определения искомых параметров. Поэтому, в данной работе рассматривается система, состоящая из трех наземных опорных источников радиоизлучения.

Структура возможных решений системы уравнений на первом этапе оказывается достаточно сложной и включает в себя большое количество вырожденных случаев. Построение эффективных алгоритмов не представляется возможным без понимания данных особенностей.

3 Локальные навигационные системы

В этой главе рассмотрены варианты локальных угломерных навигационных систем, которые позволяют определять координаты и угловую ориентацию подвижных объектов.

3.1 Локальная угломерная навигационная система первого типа

Пусть три радиоориентира расположены в точках $M_1(x_1, y_1, z_1)$, $M_2(x_2, y_2, z_2)$ и $M_3(x_3, y_3, z_3)$ с заданными координатами и находятся не на одной прямой, а ФЦ БПА, размещенной на подвижном объекте находится в точке $M_0(x, y, z)$. В таком случае, эти четыре точки образуют в пространстве треугольную пирамиду, схематическое представление которой приведено на рисунке 1. Обозначим через ℓ_i длину бокового ребра M_0M_i , через d_{ij} — длину ребра основания M_iM_j , а через α_{ij} — плоский угол $\angle M_iM_0M_j$ при вершине пирамиды. Пространственное положение подвижного объекта и радиоориентиров также будем определять с помощью радиус-векторов $\mathbf{r}_0 = (x, y, z)$ и $\mathbf{r}_i = (x_i, y_i, z_i)$ соответственно.

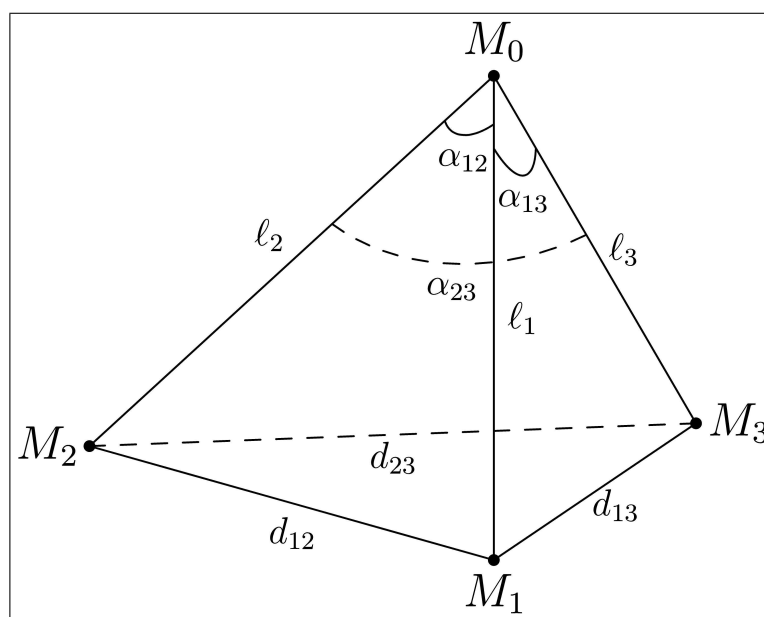


Рис. 1: Схема размещения в пространстве трех радиоориентиров и фазового центра БПА

Допустим, что в результате азимутально-угломерного радиопеленгования радиоориентиров с использованием БПА определены три пары азимутов α_i и углов места ϵ_i . Тогда в связанной системе координат БПА можно

определить три единичных вектора \mathbf{s}_{cvi} направлений на каждый из радиоориентиров следующим образом:

$$\mathbf{s}_{cvi} = (\cos \alpha_i \cos \varepsilon_i, \sin \alpha_i \cos \varepsilon_i, \sin \varepsilon_i). \quad (3.1)$$

С учетом (3.1), косинусы плоских углов α_{ij} равны

$$\cos \alpha_{ij} = (\mathbf{s}_{cvi}, \mathbf{s}_{cvj}) = \cos \varepsilon_i \cos \varepsilon_j \cos (\alpha_i - \alpha_j) + \sin \varepsilon_i \sin \varepsilon_j \quad (3.2)$$

С учетом вышеупомянутых значений, система уравнений для нахождения неизвестных значений длин ℓ_1 , ℓ_2 и ℓ_3 имеет следующий вид:

$$\begin{cases} \ell_1^2 + \ell_2^2 - 2\ell_1\ell_2 \cos \alpha_{12} = d_{12}^2 \\ \ell_1^2 + \ell_2^2 - 2\ell_1\ell_2 \cos \alpha_{12} = d_{12}^2 \\ \ell_1^2 + \ell_2^2 - 2\ell_1\ell_2 \cos \alpha_{12} = d_{12}^2 \end{cases} \quad (3.3)$$

Как показали вычислительные эксперименты, система уравнений (3.3) относительно искомым значений ℓ_1 , ℓ_2 и ℓ_3 может иметь от одного до четырех решений в каждой из областей пространства, находящихся симметрично относительно плоскости расположения трех радиоориентиров. Структура этих решений и правила отбора истинного решения будут описаны далее. Предположим, что величины ℓ_1 , ℓ_2 и ℓ_3 найдены. В таком случае, для неизвестных x , y и z точки $M_0(x, y, z)$ расположения ФЦ БПА получим следующую систему из трех уравнений

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2 = \ell_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2 = \ell_2^2 \\ (x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2 = \ell_3^2 \end{cases} \quad (3.4)$$

Для решения системы уравнений (3.4), вычтем из второго и третьего уравнений первое и перенесем члены, связанные с z в правые части уравнений, в результате чего получим линейную относительно x и y систему уравнений:

$$\begin{cases} 2x(x_1 - x_2) + 2y(y_1 - y_2) = \ell_2^2 - \ell_1^2 + x_1^2 - x_2^2 + y_1^2 - y_2^2 + z_1^2 - z_2^2 - 2z(z_1 - z_2) \\ 2x(x_1 - x_3) + 2y(y_1 - y_3) = \ell_3^2 - \ell_1^2 + x_1^2 - x_3^2 + y_1^2 - y_3^2 + z_1^2 - z_3^2 - 2z(z_1 - z_3) \end{cases} \quad (3.5)$$

Выразив, используя правило Крамера, x и y через z из системы (3.5) и подставив полученные выражения в (3.4), получим квадратное уравнение относительно z . Следует заметить, что определитель системы (3.5) не равен нулю, так как точки M_1 , M_2 и M_3 не лежат на одной прямой.

После нахождения координат точки M_0 , можно определить три единичных вектора \mathbf{s}_{hi} направлений на i -е радиоориентиров в соответствии с соотношением

$$\mathbf{s}_{hi} = \frac{\mathbf{r}_0 - \mathbf{r}_i}{|\mathbf{r}_0 - \mathbf{r}_i|}.$$

Определим квадратную матрицу \mathbf{S}_H из векторов \mathbf{s}_{H1} , \mathbf{s}_{H2} и \mathbf{s}_{H3} в соответствии с

$$\mathbf{S}_H = (\mathbf{s}_{H1}^T, \mathbf{s}_{H2}^T, \mathbf{s}_{H3}^T).$$

Также определим квадратную матрицу \mathbf{S}_{CB} из векторов (3.1):

$$\mathbf{S}_{CB} = (\mathbf{s}_{CB1}^T, \mathbf{s}_{CB2}^T, \mathbf{s}_{CB3}^T).$$

В таком случае, квадратные матрицы \mathbf{S}_H и \mathbf{S}_{CB} связаны соотношением

$$\mathbf{S}_H = \square_{CB}^H \times \mathbf{S}_{CB}, \quad (3.6)$$

где \square_{CB}^H — квадратная матрица вращения при переходе от связанной системы координат к нормальной земной системе координат. Таким образом, из (3.6) получим явное выражение для матрицы \square_{CB}^H :

$$\square_{CB}^H = \mathbf{S}_H \times \mathbf{S}_{CB}^{-1},$$

где \mathbf{S}_{CB}^{-1} — обратная к \mathbf{S}_{CB} матрица. Углы курса ψ , крена μ и тангажа ϑ находятся из матрицы \square_{CB}^H стандартным образом.

3.2 Случай локальной системы с НПУ

В работе [?] рассматривалась локальная угломерная навигационная система (ЛУНС), состоящая из трех наземных радиоориентиров и воздушного объекта, оснащенного радиолокационным оборудованием, которое способно определять азимут и угол места каждого из наземных радиоориентиров. Авторами было показано, что определение пространственных координат и угловой ориентации воздушного объекта сводится к решению нелинейной

системы уравнений следующего вида:

$$\begin{cases} \ell_1^2 + \ell_2^2 - 2\ell_1\ell_2 \cos \alpha_{12} = d_{12}^2 \\ \ell_1^2 + \ell_3^2 - 2\ell_1\ell_3 \cos \alpha_{13} = d_{13}^2 \\ \ell_2^2 + \ell_3^2 - 2\ell_2\ell_3 \cos \alpha_{23} = d_{23}^2 \end{cases}, \quad (3.7)$$

где ℓ_i — это расстояние от воздушного объекта до i -го радиоориентира, $d_{i,j}$ — расстояние между i -м и j -м наземным радиоориентиром, $\cos \alpha_{i,j}$ — плоский угол, образованный наземными ориентирами i и j и воздушным объектом. Из этой системы далее выводятся координаты и угловая ориентация воздушного объекта.

Система (3.7) может иметь несколько решений, что ведет к неоднозначности определения координат и угловой ориентации летательного аппарата. Поэтому, в статье [?] приводятся методики для расчета областей, где решение можно определить однозначно. Более того, решение системы (3.7) производится численно, что потенциально может привести к дополнительным ошибкам в определении пространственных параметров воздушного объекта. В связи с этим, далее предлагается еще один вариант ЛУНС, которая лишена вышеперечисленных недостатков ценой усложнения структуры системы.

Предлагается заменить один из пассивных радиоориентиров на наземный пункт управления (НПУ), оснащенный радиоориентиром. Требуется, чтобы НПУ мог определять азимут и угол места двух других радиоориентиров и воздушного объекта.

По аналогии с [?], будем считать, что воздушный объект находится в точке $M_0(x, y, z)$, НПУ — в $M_1(x_1, y_1, z_1)$, а i -й РО — в точке $M_i(x_i, y_i, z_i)$, $i = 1, 2$. Азимут и угол места радиоориентира M_i , полученные в результате радиопеленгования, обозначим через θ_i и β_i соответственно (см. рис. 2). Предполагается, что измерения этих углов проводятся в локальной системе координат НПУ, центр которой совпадает с координатами НПУ в НЗСК, а направления осей совпадают с НЗСК. В этой системе координат можно определить три единичных вектора направлений на радиоориентиры M_0 , M_2 и M_3 :

$$\mathbf{s}_{\text{лн}i} = (\cos \theta_i \cos \beta_i, \sin \theta_i \cos \beta_i, \sin \beta_i),$$

где $i = 0, 2, 3$. Тогда, косинусы углов $\alpha_{02} = \angle M_0 M_1 M_2$ и $\alpha_{03} = \angle M_0 M_1 M_3$

определяются как

$$\cos \alpha_{02} = (\mathbf{s}_{\text{ЛН0}}, \mathbf{s}_{\text{ЛН2}}) = \cos \beta_0 \cos \beta_2 \cos (\theta_0 - \theta_2) + \sin \beta_0 \sin \beta_2,$$

$$\cos \alpha_{03} = (\mathbf{s}_{\text{ЛН0}}, \mathbf{s}_{\text{ЛН3}}) = \cos \beta_0 \cos \beta_3 \cos (\theta_0 - \theta_3) + \sin \beta_0 \sin \beta_3.$$

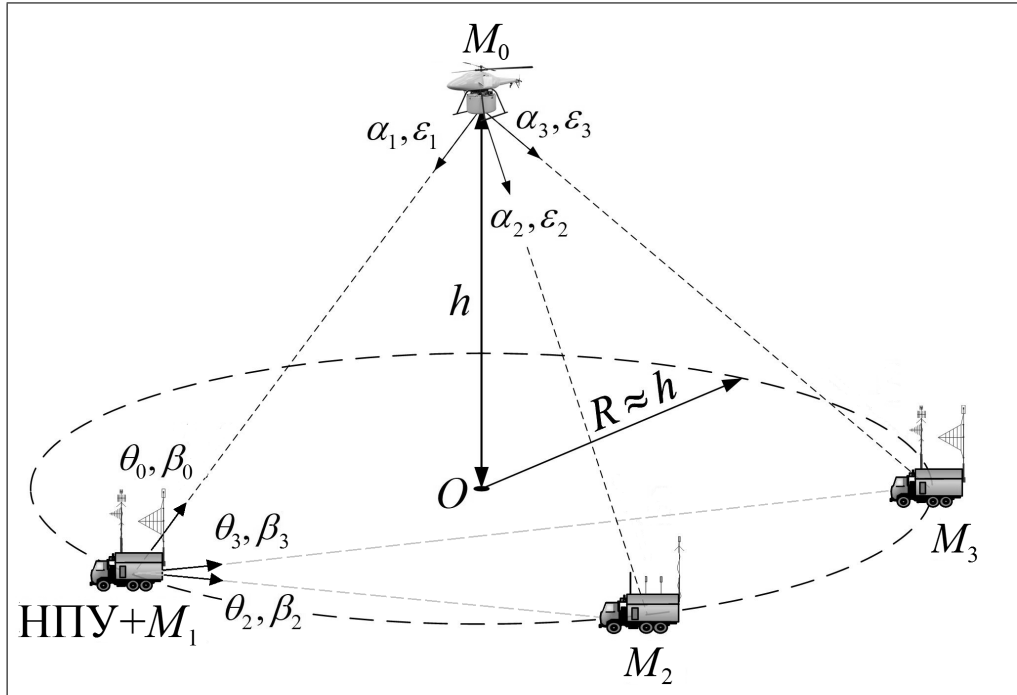


Рис. 2: Схемы размещения на местности БПЛА, НПУ и РО для реализации ЛУНС

По теореме синусов для треугольника $M_1 M_0 M_2$:

$$\frac{\ell_2}{\sin \alpha_{02}} = \frac{d_{12}}{\sin \alpha_{12}} = \frac{\ell_1}{\sin (\alpha_{12} + \alpha_{02})}. \quad (3.8)$$

Аналогично для треугольника $M_1 M_0 M_3$:

$$\frac{\ell_3}{\sin \alpha_{03}} = \frac{d_{13}}{\sin \alpha_{13}} = \frac{\ell_1}{\sin (\alpha_{13} + \alpha_{03})}. \quad (3.9)$$

Таким образом, из (3.8):

$$\ell_2 = \frac{d_{12} \sin \alpha_{02}}{\sin \alpha_{12}}.$$

В то же время, из (3.9):

$$\ell_3 = \frac{d_{13} \sin \alpha_{03}}{\sin \alpha_{13}}.$$

Длина ℓ_1 может быть найдена из любого уравнения:

$$\ell_1 = \frac{d_{12} \sin(\alpha_{12} + \alpha_{02})}{\sin \alpha_{12}} = \frac{d_{13} \sin(\alpha_{13} + \alpha_{03})}{\sin \alpha_{13}}.$$

Далее пространственные координаты и угловая ориентация БПА определяется согласно [?].

3.3 Локальная система с уменьшенным количеством наземных РО

В работе [?] было показано, что минимально возможное число наземных радиориентиров, которое позволяет однозначно определять координаты и угловую ориентацию воздушного объекта, равно трем. Однако, это число можно уменьшить, если добавить в систему еще один подвижный объект с радиориентиром и бортовой пеленгационной антенной.

Пусть подвижные объекты находятся в точках $M_1(x_1, y_1, z_1)$ и $M_2(x_2, y_2, z_2)$ а наземные ориентиры находятся в точках $M_3(x_3, y_3, z_3)$ и $M_4(x_4, y_4, z_4)$, а расстояние между ними равно d_{34} . В результате азимутально-угломестной радиопеленгации j -го радиориентира с борта i -го подвижного объекта ($i \neq j$), определяются углы азимута (α_{ij}) и угла места (ϵ_{ij}) (см рис. 3). Пусть также ℓ_{ij} — расстояние между i -м и j -м радиориентиром.

Обозначим через \mathbf{s}_{ij} единичные векторы, которые определяются следующим образом:

$$\mathbf{s}_{ij} = (\cos \alpha_{ij} \cos \epsilon_{ij}, \cos \alpha_{ij} \sin \epsilon_{ij}, \sin \epsilon_{ij}), \quad (3.10)$$

где $i \neq j, i = 1, 2, j = 1, 2, 3, 4$. Также обозначим через γ_{ij} углы, образованные воздушным объектом M_1 и радиориентирами i и j ($i \neq j$), а через φ_{ij} — углы, образованные воздушным объектом M_2 и теми же радиориентирами. В таком случае, косинусы этих углов находятся по формулам:

$$\begin{aligned} \cos \gamma_{ij} &= (\mathbf{s}_{1i}, \mathbf{s}_{1j}), i \neq j \neq 1, \\ \cos \varphi_{ij} &= (\mathbf{s}_{2i}, \mathbf{s}_{2j}), i \neq j \neq 2. \end{aligned}$$

По теореме синусов для треугольника $M_1 M_2 M_3$:

$$\frac{\ell_{12}}{\sin M_1 M_3 M_2} = \frac{\ell_{13}}{\sin \varphi_{13}} = \frac{\ell_{23}}{\sin \gamma_{13}}. \quad (3.11)$$

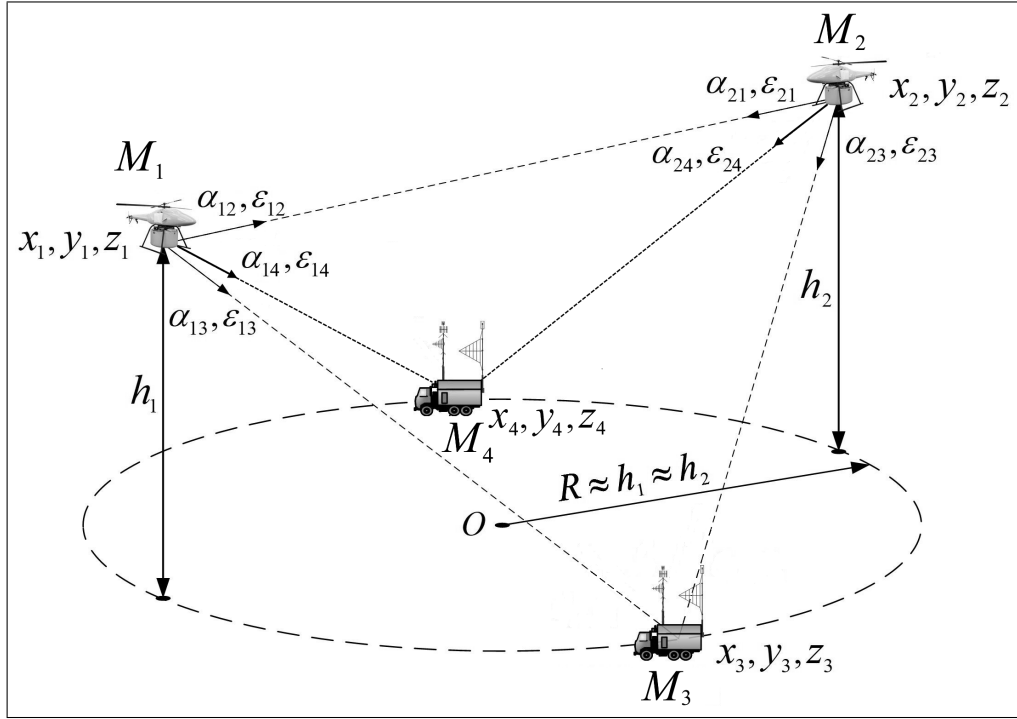


Рис. 3: Схемы размещения на местности БПЛА, НПУ и РО для реализации ЛУНС

Для треугольника $M_1 M_2 M_4$:

$$\frac{\ell_{12}}{\sin M_1 M_4 M_2} = \frac{\ell_{14}}{\sin \varphi_{14}} = \frac{\ell_{24}}{\sin \gamma_{24}}. \quad (3.12)$$

Из (3.11) и (3.12) получим отношения:

$$\frac{\ell_{23}}{\ell_{12}} = \frac{\sin \gamma_{13}}{\sin M_1 M_3 M_2}, \quad \frac{\ell_{24}}{\ell_{12}} = \frac{\sin \gamma_{14}}{\sin M_1 M_4 M_2} \quad (3.13)$$

$$\frac{\ell_{13}}{\ell_{12}} = \frac{\sin \varphi_{13}}{\sin M_1 M_3 M_2}, \quad \frac{\ell_{14}}{\ell_{12}} = \frac{\sin \varphi_{14}}{\sin M_1 M_4 M_2}. \quad (3.14)$$

По теореме косинусов для треугольника $M_2 M_4 M_3$:

$$\ell_{23}^2 + \ell_{24}^2 - 2\ell_{23}\ell_{24}\cos \varphi_{34} = \ell_{34}^2. \quad (3.15)$$

С учетом (3.13), (3.15) можно переписать в виде:

$$\ell_{12}^2 k_1^2 + \ell_{12}^2 k_2^2 - 2\ell_{12}\ell_{12}k_1k_2\cos \varphi_{34} = \ell_{34}^2, \quad (3.16)$$

где $k_1 = \sin \gamma_{13} / \sin M_1 M_3 M_2$, а $k_2 = \sin \gamma_{14} / \sin M_1 M_4 M_2$. Тогда ℓ_{12} выража-

ется следующим образом:

$$\ell_{12} = \frac{\ell_{34}}{\sqrt{k_1^2 + k_2^2 - 2k_1k_2 \cos \varphi_{34}}}. \quad (3.17)$$

С учетом (3.17), из (3.13) и (3.14) находятся расстояния ℓ_{13} , ℓ_{14} , ℓ_{23} и ℓ_{24} .

Для нахождения координат воздушного объекта M_i , введем радиус-вектор \mathbf{r}_{i3} , который определяется следующим образом:

$$\mathbf{r}_{i3} = \mathbf{s}_{i3} \ell_{i3} = \mathbf{M}_3 - \mathbf{M}_i. \quad (3.18)$$

Отсюда, координаты точки M_i в НЗСК находятся из системы уравнений:

$$\begin{cases} x_i = x_3 - \ell_{i3} \cos \alpha_{i3} \cos \varepsilon_{i3}, \\ y_i = y_3 - \ell_{i3} \cos \alpha_{i3} \sin \varepsilon_{i3}, \\ z_i = z_3 - \ell_{i3} \sin \varepsilon_{i3}. \end{cases} \quad (3.19)$$

Следует заметить, что аналогичные рассуждения справедливы и относительно радиоориентира M_4 . Совокупность расчетов координат относительно обоих наземных радиоориентиров позволяет ввести дополнительный контроль ошибок.

Отметим, что работоспособность такой конфигурации системы достигается только в случае, когда все четыре точки не лежат в одной плоскости. Это реализуется при расположении подвижных аппаратов M_1 и M_2 по разные стороны от прямой, образованной наземными радиоориентирами M_3 и M_4 (см. рис. 3).

Матрица поворота и угловая ориентация каждого из объектов находится по алгоритму, представленному в работе [?].

3.4 Случай автономной системы

Автономная угломерная радионавигационная система (АУНС) предназначена для определения координат и угловой ориентации в пространстве двух воздушных объектов, оснащенных высотомерами и бортовыми радиоориентирами с наземного пункта управления (НПУ), оснащенного радиоориентиром.

Пусть радиоориентир НПУ расположен в точке M_0 с заранее известными координатами $M_0 (x_0, y_0, z_0)$ в Балтийской системе координат (БСК), а воздушные объекты – в точках M_1 и M_2 с координатами $M_i (x_i, y_i, z_i)$, при этом z_i совпадает с измерениями высотомера h_i . НПУ M_0 способен измерять радиопеленг θ_i и угол возвышения β_i i -го воздушного объекта в БСК. Воздушные объект M_n способен измерять азимут α_{ni} и угол места ε_{ni} i -го радиоориентира (наземного или воздушного) в связанной системе координат БПА [?]. Схема размещения с указанными величинами указана на рис. 4. Пространственное положение радиоориентиров в БСК также можно охарактеризовать радиус-векторами $\mathbf{r}_i = (x_i, y_i, z_i)$, где $i = 1, 2, 3$.

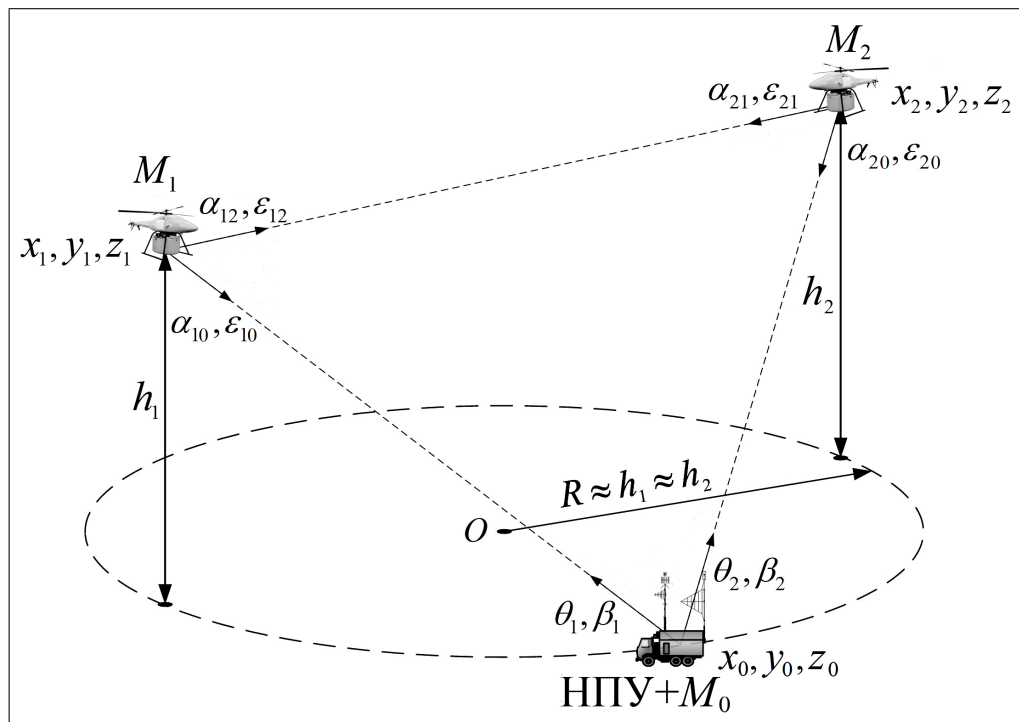


Рис. 4: Схемы размещения на местности БПЛА, НПУ и РО для реализации АУНС

При детерминированном подходе для такой системы возможно однозначно определить координаты и угловую ориентацию воздушных объектов. Для этого нужно выполнить следующие ключевые шаги:

1. Определить координаты воздушных объектов в БСК.
2. Определить координаты радиоориентиров в связанной системе координат воздушного объекта M_1 ($\Sigma_{св1}$).
3. Составить матрицу поворота системы координат $\Sigma_{св1}$ по алгоритму,

представленному ниже.

4. Определить углы поворота $\Sigma_{\text{св1}}$ по алгоритму, представленному в [?].
5. Повторить предыдущие шаги для воздушного ориентира M_2 .

Первая часть алгоритма реализуется явно — совокупность данных с высотомеров воздушных объектов и углов θ_i, β_i позволяют определить координаты летательных аппаратов однозначно. Таким образом, координаты радиоориентира M_1 и M_2 определяются следующим соотношением:

$$\begin{cases} x_1 = \rho_1 \cos \theta_1 \cos \beta_1 \\ y_1 = \rho_1 \sin \theta_1 \cos \beta_1 \\ z_1 = h_1 = \rho_1 \sin \beta_1 \\ \rho_1 = z_1 / \sin \beta_1 \end{cases}, \begin{cases} x_2 = \rho_2 \cos \theta_2 \cos \beta_2 \\ y_2 = \rho_2 \sin \theta_2 \cos \beta_2 \\ z_2 = h_2 = \rho_2 \sin \beta_2 \\ \rho_2 = z_2 / \sin \beta_2 \end{cases}. \quad (3.20)$$

Далее необходимо определить координаты радиоориентиров M_0 и M_2 в связанной системе координат воздушного объекта M_1 :

$$\begin{cases} x'_0 = \rho_{10} \cos \alpha_{10} \cos \varepsilon_{10} \\ y'_0 = \rho_{10} \sin \alpha_{10} \cos \varepsilon_{10} \\ z'_0 = z_1 - z_0 = \rho_{10} \sin \varepsilon_{10} \\ \rho_{10} = |\mathbf{r}_1 - \mathbf{r}_0| \end{cases}, \begin{cases} x'_2 = \rho_{12} \cos \alpha_{12} \cos \varepsilon_{12} \\ y'_2 = \rho_{12} \sin \alpha_{12} \cos \varepsilon_{12} \\ z'_2 = z_1 - z_2 = \rho_{12} \sin \varepsilon_{12} \\ \rho_{12} = |\mathbf{r}_1 - \mathbf{r}_2| \end{cases}, \quad (3.21)$$

где $M'_0(x'_0, y'_0, z'_0)$ и $M'_2(x'_2, y'_2, z'_2)$ — координаты M_0 и M_2 в связанной СК M_1 .

Для определения матрицы поворота системы координат $\Sigma_{\text{св1}}$, необходимо сначала ввести радиус-вектора $\mathbf{r}'_0 = (x'_0, y'_0, z'_0)$ и $\mathbf{r}'_2 = (x'_2, y'_2, z'_2)$, которые определяют положения радиоориентиров M_0 и M_2 в связанной системе координат M_1 . Далее, зададим единичные векторы $\mathbf{s}'_1, \mathbf{s}'_2$ и \mathbf{s}'_3 следующим об-

разом:

$$\begin{aligned}
\mathbf{s}'_1 &= (s'_{1x}, s'_{1y}, s'_{1z}) = (\cos \alpha_{10} \cos \varepsilon_{10}, \sin \alpha_{10} \cos \varepsilon_{10}, \sin \varepsilon_{10}), \\
\mathbf{s}'_2 &= (s'_{2x}, s'_{2y}, s'_{2z}) = (\cos \alpha_{12} \cos \varepsilon_{12}, \sin \alpha_{12} \cos \varepsilon_{12}, \sin \varepsilon_{12}), \\
\mathbf{s}'_3 &= \mathbf{s}'_1 \times \mathbf{s}'_2 = (s'_{3x}, s'_{3y}, s'_{3z}) = (\sin \alpha_{12} \cos \varepsilon_{12} \sin \varepsilon_{10} - \sin \alpha_{12} \sin \alpha_{10} \cos \varepsilon_{10}, \\
&\quad \sin \alpha_{10} \cos \varepsilon_{10} \sin \varepsilon_{12} - \cos \alpha_{10} \cos \varepsilon_{12} \sin \varepsilon_{10}, \\
&\quad \sin (\alpha_{10} - \alpha_{12}) \cos \varepsilon_{10} \cos \varepsilon_{12}).
\end{aligned} \tag{3.22}$$

Те же вектора в балтийской системе координат:

$$\begin{aligned}
\mathbf{s}_{01} &= (s_{01x}, s_{01y}, s_{01z}) = \frac{\mathbf{r}_0 - \mathbf{r}_1}{|\mathbf{r}_0 - \mathbf{r}_1|}, \\
\mathbf{s}_{21} &= (s_{21x}, s_{21y}, s_{21z}) = \frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|}, \\
\mathbf{n}_1 &= (n_{1x}, n_{1y}, n_{1z}) = \mathbf{s}_{01} \times \mathbf{s}_{21}.
\end{aligned} \tag{3.23}$$

Определим квадратную матрицу \mathbf{S} размера 3×3 координат трех полученных по формуле (3.23) единичных векторов \mathbf{s}_{01} , \mathbf{s}_{21} и \mathbf{n}_1 , записав в столбцы, в соответствии с отношением:

$$\mathbf{S}_1 = \begin{pmatrix} s_{01x} & s_{01y} & s_{01z} \\ s_{21x} & s_{21y} & s_{21z} \\ n_{1x} & n_{1y} & n_{1z} \end{pmatrix}. \tag{3.24}$$

По аналогии с (3.29) определим квадратную матрицу \mathbf{S}' размера 3×3 координат трех полученных по формуле (3.22):

$$\mathbf{S}' = \begin{pmatrix} s'_{1x} & s'_{1y} & s'_{1z} \\ s'_{2x} & s'_{2y} & s'_{2z} \\ s'_{3x} & s'_{3y} & s'_{3z} \end{pmatrix}. \tag{3.25}$$

Отсюда получим следующее преобразование координат:

$$\mathbf{R}_1 \times \mathbf{S}_1 = \mathbf{S}'$$

В таком случае, матрицу поворота связанной системы координат $\Sigma_{\text{св1}}$

можно найти следующим образом:

$$\mathbf{R}_1 = \mathbf{S}' \times \mathbf{S}_1^{-1} \quad (3.26)$$

По аналогии можно получить матрицу поворота \mathbf{R}_2 связанной системы координат M_2 . Вводятся радиус-вектора $\mathbf{r}_0'' = (x_0'', y_0'', z_0'')$, $\mathbf{r}_1'' = (x_1'', y_1'', z_1'')$, по ним же определяются единичные вектора \mathbf{s}_1'' , \mathbf{s}_2'' и \mathbf{s}_3'' :

$$\begin{aligned} \mathbf{s}_1'' &= (s_{1x}'', s_{1y}'', s_{1z}'') = (\cos \alpha_{20} \cos \varepsilon_{20}, \sin \alpha_{20} \cos \varepsilon_{20}, \sin \varepsilon_{20}), \\ \mathbf{s}_2'' &= (s_{2x}'', s_{2y}'', s_{2z}'') = (\cos \alpha_{21} \cos \varepsilon_{21}, \sin \alpha_{21} \cos \varepsilon_{21}, \sin \varepsilon_{21}), \\ \mathbf{s}_3'' &= \mathbf{s}_1'' \times \mathbf{s}_2'' = (s_{3x}'', s_{3y}'', s_{3z}'') = (\sin \alpha_{21} \cos \varepsilon_{21} \sin \varepsilon_{20} - \sin \alpha_{21} \sin \alpha_{20} \cos \varepsilon_{20}, \\ &\quad \sin \alpha_{20} \cos \varepsilon_{20} \sin \varepsilon_{21} - \cos \alpha_{20} \cos \varepsilon_{21} \sin \varepsilon_{20}, \\ &\quad \sin (\alpha_{20} - \alpha_{21}) \cos \varepsilon_{20} \cos \varepsilon_{21}). \end{aligned} \quad (3.27)$$

В балтийской системе координат:

$$\begin{aligned} \mathbf{s}_{02} &= (s_{01x}, s_{01y}, s_{01z}) = \frac{\mathbf{r}_0 - \mathbf{r}_2}{|\mathbf{r}_0 - \mathbf{r}_1|}, \\ \mathbf{s}_{21} &= (s_{21x}, s_{21y}, s_{21z}) = \frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|}, \\ \mathbf{n}_1 &= (n_{1x}, n_{1y}, n_{1z}) = \mathbf{s}_{02} \times \mathbf{s}_{21}. \end{aligned} \quad (3.28)$$

Определим квадратные матрицы \mathbf{S}_2 и \mathbf{S}'' размера 3×3 по аналогии с (3.29) и (3.25):

$$\mathbf{S}_2 = \begin{pmatrix} s_{01x} & s_{01y} & s_{01z} \\ s_{21x} & s_{21y} & s_{21z} \\ n_{1x} & n_{1y} & n_{1z} \end{pmatrix}, \mathbf{S}'' = \begin{pmatrix} s_{1x}'' & s_{1y}'' & s_{1z}'' \\ s_{2x}'' & s_{2y}'' & s_{2z}'' \\ s_{3x}'' & s_{3y}'' & s_{3z}'' \end{pmatrix}. \quad (3.29)$$

Отсюда, матрица поворота определяется следующим образом:

$$\mathbf{R}_2 = \mathbf{S}'' \times \mathbf{S}_2^{-1} \quad (3.30)$$

Углы курса ψ_1 , ψ_2 , крена μ_1 , μ_2 и тангажа ϑ_1 , ϑ_2 находятся из матриц \mathbf{R}_1 и \mathbf{R}_2 в соответствии с [?].

4 Обсуждение результатов

TODO

Заключение

TODO

Приложение А

Определение систем координат согласно ГОСТ

В данном приложении перечислены выражения, которые слишком велики для того, чтобы помещать их напрямую в текст.

А.1 Компоненты $T(I)$, $T(II)$ и $T(III)$ в сужении на M

$$\left\{ \frac{d}{dt} T(I) \Big|_{t=0} \right\} \Big|_M = \frac{1}{x_2^2} \left[Q(x_1, y_1, x_2, y_2) (-ux_2\beta_{3,1} + 2\mu x_2^3\beta_{3,2} - x_2^2\beta_{2,1} - \right. \\ \left. - x_1x_2\beta_{1,1} - x_2^2\gamma_{3,1} - x_2\sigma_{2,1} + 2vx_2y_2^2\beta_{3,2} + x_2y_1\beta_{1,2} + x_2y_2\beta_{2,2}) - \right. \\ \left. - \beta_{3,2}Q(x_1, y_1, x_2, y_2)^2 + \mu ux_2^4\beta_{3,1} + ux_2^3\gamma_{3,2} + vux_2^2y_2^2\beta_{3,1} - \mu^2x_2^6\beta_{3,2} + \mu x_2^5\beta_{2,1} + \right. \\ \left. + \mu x_1x_2^4\beta_{1,1} + \mu x_2^5\gamma_{3,1} + x_2^4\gamma_{2,2} + x_1x_2^3\gamma_{1,2} + \mu x_2^4\sigma_{2,1} + x_2^3\sigma_{3,2} - 2\mu vx_2^4y_2^2\beta_{3,2} - \mu x_2^4y_1\beta_{1,2} - \right. \\ \left. - \mu x_2^4y_2\beta_{2,2} - v^2x_2^2y_2^4\beta_{3,2} + vx_2^3y_2^2\beta_{2,1} + vx_1x_2^2y_2^2\beta_{1,1} - vx_2^2y_1y_2^2\beta_{1,2} - vx_2^2y_2^3\beta_{2,2} + \right. \\ \left. + vx_2^3y_2^2\gamma_{3,1} + x_2^3y_1\gamma_{1,1} + x_2^3y_2\gamma_{2,1} + vx_2^2y_2^2\sigma_{2,1} \right] \quad (ПА.1)$$

$$\left\{ \frac{d}{dt} T(II) \Big|_{t=0} \right\} \Big|_M = \frac{1}{x_2} F(x_1, y_1, x_2, y_2, u, v), \quad (ПА.2)$$

где $F(x_1, y_1, x_2, y_2, u, v)$ — некоторая форма четвертой степени, общий вид которой зависит от квадратичной формы $Q(x_1, x_2, y_1, y_2)$.

$$\left\{ \frac{d}{dt} T(III) \Big|_{t=0} \right\} \Big|_M = \frac{1}{x_2} \left[Q(x_1, y_1, x_2, y_2) (3\mu x_2^2\beta_{3,2} - 2vx_2y_2\beta_{3,1} + vy_2^2\beta_{3,2}) + \right. \\ \left. + 3\mu ux_2^3\beta_{3,1} + 2vux_2^2y_2\beta_{3,2} + vux_2y_2^2\beta_{3,1} - 3\mu^2x_2^5\beta_{3,2} + 3\mu x_2^4\beta_{2,1} + 3\mu x_1x_2^3\beta_{1,1} + \right. \\ \left. + 3\mu x_2^3\sigma_{2,1} + 2\mu vx_2^4y_2\beta_{3,1} - 4\mu vx_2^3y_2^2\beta_{3,2} - 3\mu x_2^3y_1\beta_{1,2} - 3\mu x_2^3y_2\beta_{2,2} + 2v^2x_2^2y_2^3\beta_{3,1} - \right. \\ \left. - v^2x_2y_2^4\beta_{3,2} + 2vx_2^3y_2\beta_{2,2} + 2vx_2^2y_1y_2\beta_{1,1} + 2vx_1x_2^2y_2\beta_{1,2} + 3vx_2^2y_2^2\beta_{2,1} + \right. \\ \left. + vx_1x_2y_2^2\beta_{1,1} - vx_2y_1y_2^2\beta_{1,2} - vx_2y_2^3\beta_{2,2} + 2vx_2^2y_2\sigma_{2,2} + vx_2y_2^2\sigma_{2,1} \right] \quad (ПА.3)$$

Приложение Б

Листинг программы

В данном приложении представлен листинги процедур, реализующие этапы представленного в пункте 3 алгоритма решения задачи. Для написания программ использовался язык Wolfram Language.

Б.1 Основные функции

```
ComposeFaces[foundation_, apex_] := Module[{foundationPairs,
  points},
foundationPairs =
  Append[Partition[foundation, 2, 1], {Last@foundation,
    First@foundation}];
points = Prepend[:, apex] & /@ foundationPairs;
points
]
```

```
PlotFigure[foundation_, apex_] := Module[{apexLow},
  (* We assume that all verices are already sorted.
  We can sort polygon in case of a
  convex shape, but it's not possible to do so otherwise *)

  apexLow = Flatten@{apex[[1 ;; 2]], 0};
Graphics3D[{EdgeForm[Thick], FaceForm[Opacity[0.9]],
  Polygon /@ ComposeFaces[foundation, apex], Point /@
    foundation,
  Point@apex,
  Point@apexLow, {Thickness[0.007], Line@{apex, apexLow}}},
  Axes -> True, AxesLabel -> {"x", "y", "z"}, FaceGrids -> All]
]
```

```
PlotPointsColored[foundation_, points_] :=
Module[{colors, colorsByLen, geomPoints, coloredPoints},
  colors = {1 -> Purple, 2 -> Blue, 3 -> LightOrange, 4 -> Orange
    ,
    5 -> Red, 6 -> Magenta};
  coloredPoints = AssignColorsToPoints[points, colors];
  Row@{Graphics3D[{{EdgeForm[Directive[Thick, Green]},
    Transparent,
    Polygon[foundation]}, coloredPoints}, Axes -> True,
```



```

    AxesLabel -> {"x", "y", "z"}, FaceGrids -> All],
Graphics3D[{{EdgeForm[Directive[Thick, Green]], Transparent,
    Polygon[foundation]}, coloredPoints}, Axes -> True,
    AxesLabel -> {"x", "y", "z"}, FaceGrids -> All,
    ViewPoint -> {0, 0, \[Infinity]}}]
]

AssignColorsToPoints[points_, colors_] :=
Module[{colorsByLen, geomPoints, coloredPoints},
    colorsByLen = Length /@ points /. colors;
    geomPoints = Map[Point, points, {2}];
    coloredPoints =
        Table[{colorsByLen[[i]], geomPoints[[i]]}, {i, Length@points
            }];
    coloredPoints
]

ExtractApex[foundation_, sides_, startEq_: 0, endEq_: 0] :=
Module[{system, point, start, end},
    If [endEq == 0, end = Length@foundation];
    If[startEq == 0, start = 1];
    system =
        Table[(foundation[[i, 1]] - x)^2 + (foundation[[i, 2]] -
            y)^2 + (foundation[[i, 3]] - z)^2 == Subscript[1, i]^2,
            {i,
                start, end}] /. sides;
    system = Append[system, z >= 0];
    point = NSolve[system, {x, y, z}];
    point
]

SolveForThree[triangle_, apex_] :=
Module[{system, \[Alpha], d, ll, solution},
    (* Prepare foundation distances *)

    Subscript[d, 1, 2] =
        SquaredEuclideanDistance[triangle[[1]], triangle[[2]]];
    Subscript[d, 2, 3] =
        SquaredEuclideanDistance[triangle[[2]], triangle[[3]]];
    Subscript[d, 3, 1] =
        SquaredEuclideanDistance[triangle[[3]], triangle[[1]]];
    (* Prepare distances to vertex *)

```

```

11 = Table[SquaredEuclideanDistance[triangle[[i]], apex], {i,
3}];
(* Calculate flat angles *)

Subscript[\[Alpha], 1,
2] = -((Subscript[d, 1, 2] - 11[[1]] - 11[[2]])/(
2*Sqrt@11[[1]]*Sqrt@11[[2]]));
Subscript[\[Alpha], 2,
3] = -((Subscript[d, 2, 3] - 11[[2]] - 11[[3]])/(
2*Sqrt@11[[2]]*Sqrt@11[[3]]));
Subscript[\[Alpha], 3,
1] = -((Subscript[d, 3, 1] - 11[[3]] - 11[[1]])/(
2*Sqrt@11[[3]]*Sqrt@11[[1]]));
system = {
Subscript[1, 1]^2 + Subscript[1, 2]^2 -
2 * Subscript[1, 1] * Subscript[1, 2]*Subscript[\[Alpha],
1,
2] == Subscript[d, 1, 2],
Subscript[1, 2]^2 + Subscript[1, 3]^2 -
2 * Subscript[1, 2] * Subscript[1, 3]*Subscript[\[Alpha],
2,
3] == Subscript[d, 2, 3],
Subscript[1, 3]^2 + Subscript[1, 1]^2 -
2 * Subscript[1, 3] * Subscript[1, 1]*Subscript[\[Alpha],
3,
1] == Subscript[d, 3, 1],
Subscript[1, 1] >= 0, Subscript[1, 2] >= 0, Subscript[1, 3]
>= 0
};
solution =
NSolve[system, {Subscript[1, 1], Subscript[1, 2], Subscript[1,
3]}, Reals, Method -> "Newton"];
Flatten[{x, y, z} /. # & /@ ExtractApex[triangle, #] & /@
solution,
1]
]

SolveForFour4[quad_, apex_, prevRoot_: {}] :=
Module[{system, \[Alpha], d, 11, solution, sol13, sol24},
(* Prepare foundation distances *)

```

```

Subscript[d, 1, 2] =
  SquaredEuclideanDistance[quad[[1]], quad[[2]]];
Subscript[d, 2, 3] =
  SquaredEuclideanDistance[quad[[2]], quad[[3]]];
Subscript[d, 3, 4] =
  SquaredEuclideanDistance[quad[[3]], quad[[4]]];
Subscript[d, 4, 1] =
  SquaredEuclideanDistance[quad[[4]], quad[[1]]];
(* Prepare distances to vertex *)

l1 = Table[SquaredEuclideanDistance[quad[[i]], apex], {i, 4}];
(* Calculate flat angles *)

Subscript[\[Alpha], 1,
  2] = -((Subscript[d, 1, 2] - l1[[1]] - l1[[2]])/(
  2*Sqrt@l1[[1]]*Sqrt@l1[[2]]));
Subscript[\[Alpha], 2,
  3] = -((Subscript[d, 2, 3] - l1[[2]] - l1[[3]])/(
  2*Sqrt@l1[[2]]*Sqrt@l1[[3]]));
Subscript[\[Alpha], 3,
  4] = -((Subscript[d, 3, 4] - l1[[3]] - l1[[4]])/(
  2*Sqrt@l1[[3]]*Sqrt@l1[[4]]));
Subscript[\[Alpha], 4,
  1] = -((Subscript[d, 4, 1] - l1[[4]] - l1[[1]])/(
  2*Sqrt@l1[[4]]*Sqrt@l1[[1]]));

system = {
  Subscript[1, 1]^2 + Subscript[1, 2]^2 -
    2 * Subscript[1, 1] * Subscript[1, 2]*Subscript[\[Alpha],
    1,
    2] == Subscript[d, 1, 2],
  Subscript[1, 2]^2 + Subscript[1, 3]^2 -
    2 * Subscript[1, 2] * Subscript[1, 3]*Subscript[\[Alpha],
    2,
    3] == Subscript[d, 2, 3],
  Subscript[1, 3]^2 + Subscript[1, 4]^2 -
    2 * Subscript[1, 3] * Subscript[1, 4]*Subscript[\[Alpha],
    3,
    4] == Subscript[d, 3, 4],
  Subscript[1, 4]^2 + Subscript[1, 1]^2 -
    2 * Subscript[1, 4] * Subscript[1, 1]*Subscript[\[Alpha],
    4,

```

```

1] == Subscript[d, 4, 1],
Subscript[1, 1] >= 0, Subscript[1, 2] >= 0, Subscript[1, 3]
    >= 0,
Subscript[1, 4] >= 0
};
solution =
NSolve[system, {Subscript[1, 1], Subscript[1, 2], Subscript[1,
    3],
    Subscript[1, 4]}, Reals];
sol13 =
Flatten[{x, y, z} /. # & /@ ExtractApex[quad, #, 1, 3] & /@
    solution, 1];
sol24 =
Flatten[{x, y, z} /. # & /@ ExtractApex[quad, #, 2, 4] & /@
    solution, 1];
Intersection[sol13, sol24,
    SameTest -> (EuclideanDistance[#1, #2] < 10(-5)) &)]
]
SolveForFour6[quad_, apex_] :=
Module[{system, \[Alpha], d, ll, solution, sol13, sol24},
    (* Prepare foundation distances *)

    Subscript[d, 1, 2] =
        SquaredEuclideanDistance[quad[[1]], quad[[2]]];
    Subscript[d, 2, 3] =
        SquaredEuclideanDistance[quad[[2]], quad[[3]]];
    Subscript[d, 3, 4] =
        SquaredEuclideanDistance[quad[[3]], quad[[4]]];
    Subscript[d, 4, 1] =
        SquaredEuclideanDistance[quad[[4]], quad[[1]]];
    Subscript[d, 1, 3] =
        SquaredEuclideanDistance[quad[[1]], quad[[3]]];
    Subscript[d, 2, 4] =
        SquaredEuclideanDistance[quad[[2]], quad[[4]]];
    (* Prepare distances to vertex *)

    ll = Table[SquaredEuclideanDistance[quad[[i]], apex], {i, 4}];
    (* Calculate flat angles *)

    Subscript[\[Alpha], 1,
        2] = -((Subscript[d, 1, 2] - ll[[1]] - ll[[2]])/(
            2*Sqrt@ll[[1]]*Sqrt@ll[[2]]));

```

```

Subscript[\[Alpha], 2,
  3] = -((Subscript[d, 2, 3] - l1[[2]] - l1[[3]])/(
    2*Sqrt@l1[[2]]*Sqrt@l1[[3]]));
Subscript[\[Alpha], 3,
  4] = -((Subscript[d, 3, 4] - l1[[3]] - l1[[4]])/(
    2*Sqrt@l1[[3]]*Sqrt@l1[[4]]));
Subscript[\[Alpha], 4,
  1] = -((Subscript[d, 4, 1] - l1[[4]] - l1[[1]])/(
    2*Sqrt@l1[[4]]*Sqrt@l1[[1]]));
Subscript[\[Alpha], 1,
  3] = -((Subscript[d, 1, 3] - l1[[1]] - l1[[3]])/(
    2*Sqrt@l1[[1]]*Sqrt@l1[[3]]));
Subscript[\[Alpha], 2,
  4] = -((Subscript[d, 2, 4] - l1[[2]] - l1[[4]])/(
    2*Sqrt@l1[[2]]*Sqrt@l1[[4]]));

system = {
  Subscript[1, 1]^2 + Subscript[1, 2]^2 -
    2 * Subscript[1, 1] * Subscript[1, 2]*Subscript[\[Alpha],
    1,
    2] == Subscript[d, 1, 2],
  Subscript[1, 2]^2 + Subscript[1, 3]^2 -
    2 * Subscript[1, 2] * Subscript[1, 3]*Subscript[\[Alpha],
    2,
    3] == Subscript[d, 2, 3],
  Subscript[1, 3]^2 + Subscript[1, 4]^2 -
    2 * Subscript[1, 3] * Subscript[1, 4]*Subscript[\[Alpha],
    3,
    4] == Subscript[d, 3, 4],
  Subscript[1, 4]^2 + Subscript[1, 1]^2 -
    2 * Subscript[1, 4] * Subscript[1, 1]*Subscript[\[Alpha],
    4,
    1] == Subscript[d, 4, 1],
  Subscript[1, 1]^2 + Subscript[1, 3]^2 -
    2 * Subscript[1, 1] * Subscript[1, 3]*Subscript[\[Alpha],
    1,
    3] == Subscript[d, 1, 3],
  Subscript[1, 2]^2 + Subscript[1, 4]^2 -
    2 * Subscript[1, 2] * Subscript[1, 4]*Subscript[\[Alpha],
    2,
    4] == Subscript[d, 2, 4],

```

```

    Subscript[1, 1] >= 0, Subscript[1, 2] >= 0, Subscript[1, 3]
      >= 0,
    Subscript[1, 4] >= 0
  };
solution =
  Solve[system, {Subscript[1, 1], Subscript[1, 2], Subscript[1,
    3],
    Subscript[1, 4]}, Reals];
sol13 =
  Flatten[{x, y, z} /. # & /@ ExtractApex[quad, #, 1, 3] & /@
    solution, 1];
sol24 =
  Flatten[{x, y, z} /. # & /@ ExtractApex[quad, #, 2, 4] & /@
    solution, 1];
Intersection[sol13, sol24,
  SameTest -> (EuclideanDistance[#1, #2] < 10^(-5) &)]
]

```

Б.2 Пример использования функций

Листинг программы для нахождения координат подвижного объекта на различных высотах в случае, когда РО расположены в вершинах правильного треугольника, а подвижный объект начинает подъем из центра масс треугольника:

```

M1 = {0, 0, 0};
M2 = {6, 9, 0};
M3 = {10, 0, 0};
M0 = {5, 4, 6};
foundation = {M1, M2, M3};
PlotFigure[foundation, M0]
solutions =
  Table[{M0[[1]], M0[[2]], i},
    SolveForThree[
      foundation, {M0[[1]], M0[[2]],
        i}], {i, 0, 10, 0.1}];
Grid@solutions;
PlotPointsColored[foundation, solutions[[1 ;;, 2]]]

```