

Ставропольский край,
Андроповский муниципальный округ,
Муниципальное бюджетное общеобразовательное учреждение «Средняя
общеобразовательная школа №11
имени Ю.В.Андропова»
**Муниципальный этап краевого конкурса-выставки
научно-технического творчества молодежи «Таланты XXI века»**

Программирование группы квадрокоптеров

Автор:

Центр «Точка роста»

«Геоинформационные технологии»

Светлова Айсари Магомедовна.

Научный руководитель:

Учитель, педагог центра «Точка роста»

Белашова Татьяна Владимировна.

с.Солуно-Дмитриевское,

2022 год.

Оглавление:

Аннотация.....	3
Введение	4
Обзор объекта управления. Мультикоптеры	3
Постановка задачи	4
Алгоритм управления группой квадрокоптеров	5
Взлёт. Алгоритм взлёта	6
Полёт по заданной траектории. Алгоритм облёта территории	7
Посадка. Алгоритм посадки	8
Реализация алгоритма и результаты моделирования	8
Блок-схема тела программы	9
Полётные характеристики и эксперимент	9
Заключение	10
Список источников	11
Приложение.....	12-14

Аннотация

Недавно в нашей школе развернули Центр дополнительного образования «Точка роста» (см. на слайд 2). На занятиях кружка «Геоинформационные системы» мы с одноклассниками впервые познакомились с квадрокоптерами DJI Mavic Air и DJI Tello, их ещё называют дронами. Мы учимся программировать дроны.



Исследуя различные области применения дронов (см. на слайд 3), мы решили попробовать научить группу дронов (в нашем случае это дроны Tello и квадрокоптер DJI Mavic Air) перемещаться по определённым траекториям. Задачи, которые могут решать небольшие группы дронов могут быть разного рода. Начиная от доставки почты и до аэросъёмки сельхозугодий. Во всех этих случаях можно запрограммировать дроны, так чтобы они следовали, как в группе так и отдельно, по определённым маршрутам.

Цель нашего исследования - разработка алгоритма управления группой квадрокоптеров. Мы создали модель группового управления, на основе которого написан алгоритм, реализующий выполнение групповой задачи.

В процессе работы мы пришли к выводу, что использование группы дронов с их способностью коллективно выполнять сложные задачи, многократно увеличивает эффективность той или иной работы. По нашим данным, только доставка почты может сэкономить до 500 тыс. рублей в год в пределах

одного населенного пункта нашего района. Эти средства можно направить на приобретение нескольких дронов и оплату работы специалиста.

Ключевые слова: квадрокоптер, группа, централизованное управление, алгоритм управления, дрон («Дрон» (англ. *drone* — **трутень**) — беспилотный летательный аппарат или БПЛА).

Введение

На данный момент большинство людей воспринимают квадрокоптер больше как игрушку, нежели как беспилотный летательный аппарат. Технологии, которые используются в квадрокоптерах, такие как бортовые компьютеры, электродвигатели, аккумуляторы и в главную очередь программное обеспечение, очень быстро развиваются. Современными квадрокоптерами, находящимися до часу в воздухе, легко маневрирующими между препятствиями и поднимающими тяжелое оборудование может управлять даже ребенок.

Квадрокоптеры под управлением оператора могут выполнять задачи различного рода сложности, будь то это съемка материалов для фильма или поисковая операция. Однако наибольшую ценность представляет возможность автоматизации выполняемой задачи, когда требуется только один оператор, который следит за четким выполнением работы. Можно автоматизировать процесс доставки грузов, сканирования местности, а также поисково-спасательной операции.

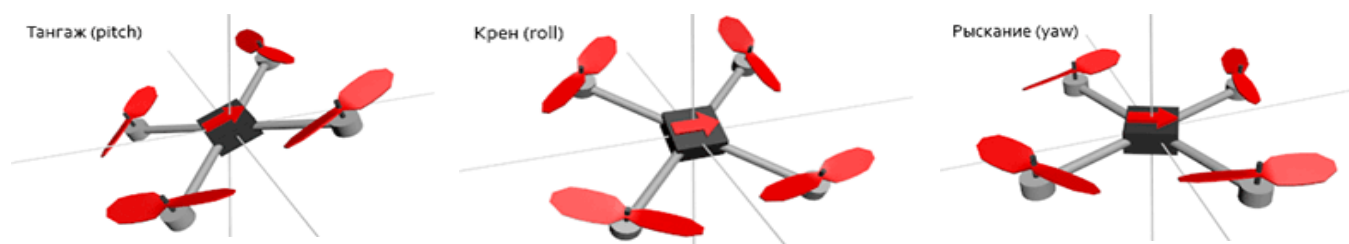
Группа квадрокоптеров имеет преимущества относительно одиночного квадрокоптера, такие как: взаимозаменяемость в случаях внештатных ситуаций, расширенная информированность об окружающей среде за счет коммуникации внутри группы, повышенная надежность и скорость выполнения задач. В данной работе рассматривается задача автоматизации работы группы квадрокоптеров на местности, в которой дроны совершают взлет, посадку и полет по заданной траектории.

Обзор объекта управления. Мультикоптеры

Мультикоптер (англ. multicopter, multicopter, многороторный вертолёт, многолёт) — это летательный аппарат, построенный по вертолётной схеме с тремя и более несущими винтами. Многовинтовые вертолёты разрабатывались ещё в первые годы вертолётостроения. Один из первых квадрокоптеров (англ. quadcopter, четырёхроторный вертолёт), который реально оторвался от земли и мог держаться в воздухе, был создан Георгием Ботезатом и испытан в 1922 году. Новые разработки начались в 1950-е годы, но дальше прототипов дело не продвинулось. Новое рождение мультикоптеры получили в XXI веке, уже как беспилотные аппараты. Благодаря простоте конструкции квадрокоптеры часто используются в любительском моделировании.

Мультикоптеры удобны для недорогой аэрофото- и киносъёмки — громоздкая камера вынесена из зоны действия винтов. Мультикоптеры имеют 3 или более винтов. Каждый винт приводится в движение собственным двигателем. Половина винтов вращается по часовой стрелке, половина — против, поэтому хвостовой винт мультикоптеру не нужен. Маневрируют мультикоптеры путём изменения скорости вращения винтов. Например: ускорить все винты — *подъём*; ускорить винты с одной стороны и замедлить с другой — *движение в сторону*; ускорить винты, вращающиеся по часовой стрелке, и замедлить вращающиеся против — *поворот*.

Углы тангажа, крена и рыскания (pitch, roll, yaw) — углы, которыми принято определять и задавать ориентацию квадрокоптера в пространстве.



Уровень автономности квадрокоптера бывает разным. При некоторых задачах управление квадрокоптером может вестись оператором с наземной рабочей станции. В этом случае квадрокоптер находится в зоне видимости оператора и/или изображение с бортовой камеры транслируется на экран рабочей станции, оператор при этом может задавать маршрут и траекторию движения летательного аппарата. Оперативное управление квадрокоптером по скорости может производиться джойстиком радиопульта, применяя простые команды: «вверх», «вниз», «направо», «налево».

В общем случае составными частями квадрокоптера являются механическая часть (рама, шасси, пропеллеры), плата управления, блок аккумуляторов, электроприводы, контроллеры приводов и различные датчики, в том числе камера.

Постановка задачи

Задачей данной работы является построение модели, создание и написание алгоритма, который позволит группе квадрокоптеров совершать одновременную работу, а также обеспечивать взаимосвязь друг с другом.

Необходимость группового управления обусловлена тем, что многие задачи выполняются быстрее, точнее и с меньшими тратами ресурсов. Преимуществами данного типа управления являются: больший радиус действия, расширенный набор выполняемых функций и более высокая вероятность выполнения задания, достигаемая за счет возможности перераспределения целей между роботами группы в случае выхода из строя некоторых из них.

Группа квадрокоптеров с децентрализованным управлением может быть применена для решения ряда практических задач в сельском хозяйстве. На практике мультикоптеры среди БПЛА оказываются более востребованными, поскольку их можно использовать и для других задач, например, для слежения за сельскохозяйственной техникой в полях, где требуется способность зависания в воздухе. Недостатком использования мультикоптеров в задаче картирования сельскохозяйственных площадей является ограниченность времени и расстояния их полета. Покрытие масштабных площадей возможно при использовании группы квадрокоптеров, управляемой оператором как единое целое.

Алгоритм управления группой квадрокоптеров

В этом разделе мы смоделировали управление группой квадрокоптеров компьютерной программой. Необходимо разработать сценарий полета, алгоритмы взлета и посадки, алгоритмы поведения в воздухе и избегания столкновений. Ключевую роль в программировании квадрокоптера занимают лётные характеристики выбранного квадрокоптера.

Алгоритм работы группы квадрокоптеров представляет собой совокупность множества сценариев: последовательность выхода на рабочую траекторию,

выход на траекторию, выполнение работы, посадка. Все они будут рассмотрены ниже.

Взлёт. Алгоритм взлёта

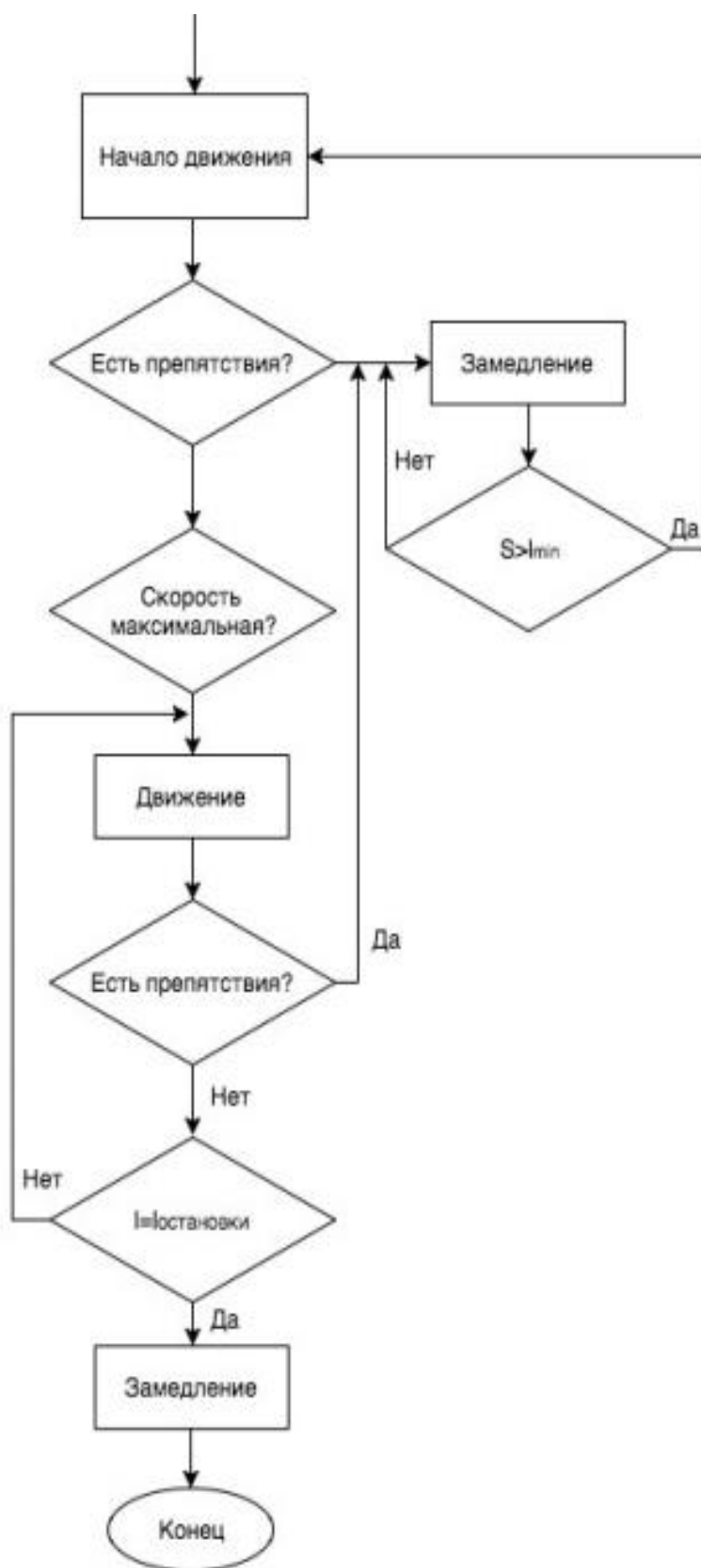
После выбора оптимального квадрокоптера наблюдатель посылает ему команду на взлёт. После этого дрон начинает взлёт.



z – высота дрона в «реале», h – заданная оператором высота.

Полёт по заданной траектории. Алгоритм облёта траектории

На данном этапе квадрокоптер должен облететь траекторию и вернуться в точку, с которой он начинал движение.



Посадка. Алгоритм посадки

Посадка квадрокоптера происходит аналогично взлету, однако при дополнительных условиях: дрон, выходящий на посадку имеет преимущество над всеми остальными, в связи с этим ему не требуется замедление или обработка столкновений.



Реализация алгоритма и результаты моделирования

Реализацию алгоритма логично разбить на несколько составляющих для того чтобы можно было менять реализацию составных элементов без риска повредить весь функционал программы. Также это необходимо для того, чтобы программа могла быть реализована не целиком на каждом устройстве, а только та ее часть, которая к нему относится. Тело программы (функция main) будет выглядеть следующим образом:

1. Задать размеры поля и количество квадрокоптеров;
2. Создание поля и экземпляров квадрокоптеров;
3. Взлет квадрокоптера;
4. Отработка траектории;
5. Посадка квадрокоптера и взлет другого квадрокоптера;

6. Повтор пунктов 2-5 до остановки программы.

Блок-схема тела программы



Полётные характеристики и эксперимент

Параметры моделируемой системы выглядят следующим образом:

$n = 3$ — количество квадрокоптеров, одновременно выполняющих задачу.

$T(\text{полета}) = 5$ минут — время, которое 1 квадрокоптер может находиться в воздухе.

$T(\text{посадки}) = 1$ минута — время, за которое 1 квадрокоптер садится на платформу.

$T(\text{подъема}) = 1$ минута — время подъема одного квадрокоптера на рабочую высоту.

$R(\text{безопасный}) = 20$ см — радиус безопасной сферической области вокруг квадрокоптера.

$S(\text{площадки}) = 20\text{см} \times 20\text{ см}$ — площадь площадки, на которую может сесть квадрокоптер.

$T(\text{зарядки}) = 7\text{ мин.}$ – время полной зарядки одного квадрокоптера.

$T(\text{выхода}) = 0\text{ сек}$ — время выхода квадрокоптера на рабочую траекторию, либо в точку посадки.

$T(\text{работы}) = T(\text{полета}) - (T(\text{подъема}) + T(\text{посадки}) + 2 \cdot T(\text{выхода}))$ — время полезной работы, выполняемой квадрокоптером.

$L(\text{траектории})$ — длина траектории, по которой летит один квадрокоптер.

Важно подобрать такую длину траектории, при которой выполняющий работу квадрокоптер не будет полностью разряжен, а так же вернется в точку взлета.

На основе рассмотренных алгоритмов была написана программа на языке Python, которая моделирует поведение квадрокоптеров в группе.

Заключение

В ходе работы над проектом нами разработан алгоритм управления группы квадрокоптеров. На основе этого алгоритма была реализована программа на языке Python(см.Приложение).

выполняющая роль наблюдателя, который распределяет время вылета, подает команды на взлет, посадку и работу по траектории дронов.

Проведение численного эксперимента показало, что модель отрабатывает все заданные команды. В дальнейшем планируется усложнение алгоритма, для реализации более гибкого управления объектами, движению по сложным траекториям, возможностью изменения траектории в ходе работы и обработке большего количества внештатных ситуаций.

Список источников

1. <https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BA%D0%BE%D0%BF%D1%82%D0%B5%D1%80>
2. <http://kvadrokopters.com/blog/interesnoe/kak-rabotaet-kvadrokopter/>
3. <https://nashol.com/2014072079020/modeli-i-algoritmi-kollektivnogo-upravleniya-v-gruppah-robotov-kalyaev-i-a-gaiduk-a-r-kapustyan-s-g.html>
4. <https://habr.com/ru/post/227425/>
5. <https://cyberleninka.ru/article/v/upravlenie-dvizheniem-kvadrokoptera-po-zaranee-zadannoy-traektorii>
6. <https://habr.com/ru/post/281591/>

Приложение

Программа на языке Python

```
#!/usr/bin/python
```

```
#!/*- coding: utf-8 -*-
```

```
import time
```

```
from dronekit import connect, VehicleMode, LocationGlobalRelative
```

```
# Подключаемся
```

```
vehicle = connect('unpin:10.1.1.10:14550', wait_ready=True)
```

```
def arm_and_takeoff(aTargetAltitude):
```

```
    """ Запускае двигатели и взлетаем до высоты aTargetAltitude  
    """
```

```
    print "Предполетные проверки"
```

```
    while not vehicle.is_armable:
```

```
        print "Ждем коптер..."
```

```
        time.sleep(1)
```

```
    print "Запускаем двигатели"
```

```
    # Устанавливаем GUIDED режим
```

```
    vehicle.mode = VehicleMode("GUIDED")
```

```
    # Запускаем моторы
```

```
    vehicle.armed = True
```

```
    # Ждем пока моторы раскрутятся.
```

```
    # Поскольку все происходит в реальном мире, нужно не забывать что на  
    выполнение действий требуется время.
```

```
    while not vehicle.armed:
```

```
        print " Ждем моторы..."
```

```
        time.sleep(1)
```

```
    print "Взлет!"
```

```
# Влетаем до нужной высоты
vehicle.simple_takeoff(aTargetAltitude)

# Даем задание подняться до нужной высоты
# и... ждем пока поднимемся.

while True:
    print " Текущая высота: ", vehicle.location.global_relative_frame.alt
    # Смотрим как высоко уже поднялись
    if vehicle.location.global_relative_frame.alt>=aTargetAltitude*0.95:
        print "Поднялись на %d метров" % vehicle.location.global_relative_frame.alt
        break
    time.sleep(1)
    print "Достигли заданной высоты"

# Запускаемся и поднимаемся на 20 метров
arm_and_takeoff(20)

# Задаем координаты нужной точки
a_location = LocationGlobalRelative(-43.1944, -46.9007, 20)
# полетели
vehicle.simple_goto(a_location)

# Путевая скорость, м/с
vehicle.airspeed = 7.5

def is_arrived(lat, lon, alt, precision=0.3):
    # текущая позиция
    veh_loc = vehicle.location.global_relative_frame
    # получаем данные в метрах
    diff_lat_m = (lat - veh_loc.lat) * 1.113195e5
    diff_lon_m = (lon - veh_loc.lon) * 1.113195e5
```

```
diff_alt_m = alt - veh_loc.alt
```

```
# отклонение
```

```
dist_xyz = math.sqrt(diff_lat_m**2 + diff_lon_m**2 + diff_alt_m**2)
```

```
if dist_xyz < precision:
```

```
    print "Прибыли на место"
```

```
    return True
```

```
else:
```

```
    print "Еще не долетели"
```

```
    return False
```

```
# проверяем долетели или еще нет
```

```
while not is_arrived(lat, lon, alt):
```

```
# ждем еще 3 секунды
```

```
time.sleep(3)
```

```
# прилетели на место и сделали что-нибудь полезное
```

```
time.sleep(10)
```

```
print "Возвращаемся"
```

```
vehicle.mode = VehicleMode("RTL")
```

```
:
```