

UAVPROF DRONE SIMULATOR

Competition

Инструкция по эксплуатации

Настоящая документация может быть использована только для поддержки работоспособности продуктов, установленных на основании договора с ООО «СТРАТУС». Документация может быть передана на основании договора, по которому производится (производилась или будет производиться) установка продуктов, или явно выраженного согласия ООО «СТРАТУС» на использование данной документации. Если данный экземпляр документации попал к вам каким-либо иным образом, пожалуйста, сообщите об этом в ООО «СТРАТУС» по адресу, приведенному ниже.

Все примеры, приведенные в документации (в том числе примеры отчетов и экранных форм), составлены на основании тестовой базы ООО «СТРАТУС». Любое совпадение имен, фамилий, названий компаний, банковских реквизитов и другой информации с реальными данными является случайным.

Все встречающиеся в тексте торговые знаки и зарегистрированные торговые знаки являются собственностью их владельцев и использованы исключительно для идентификации программного обеспечения или компаний.

Данная документация может не отражать некоторых модификаций программного обеспечения. Если вы заметили в документации ошибки или опечатки или предполагаете их наличие, пожалуйста, сообщите об этом в ООО «СТРАТУС».

Все имущественные авторские права сохраняются за ООО «СТРАТУС» в соответствии с действующим законодательством.

© ООО «СТРАТУС», 2024

ООО «СТРАТУС»

119607, г. Москва, вн.тер.г. Муниципальный округ Раменки, б-р Раменский, д. 1

Тел.: +7 (931) 604-34-33

Электронная почта: sim@uavprof.com

Содержание

1. Общие сведения	3
1.1. Модели аппаратов	3
1.1.1. Общая информация для всех аппаратов	3
1.1.2. Камера	4
1.2. Системы координат	4
1.3. Алгоритм управления	4
1.3.1. Управляющие воздействия	4
1.3.2. Изменение коэффициентов ПИД-регуляторов автопилота	5
1.4. Формации	5
1.5. Использование камеры аппарата	5
1.6. Горячие клавиши переключения видов	6
1.7. Переключение на Vulkan	6
2. Начало работы	7
3. Дисциплины	8
3.1. Синхронный полёт	8
3.1.1. Гоночная трасса	8
3.1.2. Формации	9
3.1.3. Задание	10
3.1.4. Запуск	11
3.1.5. Критерии оценивания	11
3.2. Командная гонка	12
3.2.1. Гоночная трасса	13
3.2.2. Аппарат	14
3.2.3. Задание	14
3.2.4. Запуск	14
3.2.5. Критерии оценивания	14
3.3. Уход от столкновения	15
3.3.1. Гоночная трасса	15
3.3.2. Задание	16
3.3.3. Запуск	17
3.3.4. Критерии оценивания	18
4. Судейство	19
История изменений	20

1. Общие сведения

Продукт UAVPROF Drone Simulator: Competition — программа имитации полётов (далее — Симулятор) для обучения и отработки навыков управления беспилотным воздушным судном (далее — БВС) с использованием программного обеспечения, создаваемого пользователем в процессе работы с продуктом.

Продукт предназначен для проведения соревнований и позволяет использовать его для двух дивизионов с разным уровнем подготовки — студенческого и профессионального.

Для запуска продукта используется командная строка, а ПО участников подключается к автопилоту по сети через MAVLink по UDP и к симулятору через ROS 2.

Для судейства в поставку продукта входят файлы с параметрами оценивания (подробнее см. в разделе «[Судейские файлы](#)»).

1.1. Модели аппаратов

В зависимости от дисциплины и дивизиона, используется одна или группа из следующих моделей БПЛА мультироторного типа:

- FPV F5D Auton;
- FPV F5D Auton GPS (доступна GPS);
- FPV F5D Auton LPS (доступна LPS — система локального позиционирования).

Наборы параметров автопилотов указаны в файлах, расположенных в директории params соответствующей версии Симулятора (<https://gitflic.ru/project/uavprofsim/parma-bas/file?file=docs%2Fparams>):

- fpv_f5d_auton.params — для FPV F5D Auton;
- fpv_f5d_auton_gps.params — для FPV F5D Auton GPS;
- fpv_f5d_auton_lps.params — для FPV F5D Auton LPS.

Описание параметров автопилота см. https://docs.px4.io/v1.13/en/advanced_config/parameter_reference.html.

1.1.1. Общая информация для всех аппаратов

- Масса: 0.696 кг;
- размер: 0.263 x 0.316 x 0.089 м;
- доступна [камера](#);
- PX4 1.13.2.

Табл. 1. Координаты пропеллеров относительно центра масс

№	x, м	y, м	z, м
1	0.068	-0.094	0.010
2	-0.068	0.082	0.010
3	0.068	0.094	0.010
4	-0.068	-0.082	0.010

1.1.2. Камера

На всех моделях аппаратов доступна камера со следующими характеристиками:

- цветное изображение (RGB);
- вертикальный угол обзора (VFOV) — 86.8 градусов;
- разрешение 1280x720 пикселей;
- максимальное количество кадров в секунду (FPS) – 60;
- изображения передаются через топик ROS 2;
- координаты относительно центра масс: (0.107 м, 0.000 м, 0.000 м);
- угол наклона (тангаж): 20 градусов.

1.2. Системы координат

Используются следующие системы координат:

- локальная система координат автопилота;
- локальная система координат Симулятора;
- глобальная система координат ГНСС.

Базисы, соответствующие осям X, Y, Z локальных систем координат:

- ENU (Восток-Север-Вверх) для локальной системы координат Симулятора, сообщений ROS/MAVROS;
- NED (Север-Восток-Вниз) для автопилота PX4 и сообщений MAVLink.

Точке отсчета (0, 0, 0) локальной системы координат Симулятора соответствует точка (Широта, Долгота, Высота) глобальной системы координат, задаваемая в JSON-файле параметров (свойства `latitude`, `longitude`, `altitude` объекта `/session/world`). Все координаты объектов в текущем документе заданы в базисе ENU (Восток-Север-Вверх) локальной системы координат Симулятора.

1.3. Алгоритм управления

Управление каждым аппаратом в группе происходит с помощью алгоритма, разрабатываемого Командой.

Управляющие воздействия на аппараты выдаются в одном из следующих вариантов:

- по точкам в локальной системе координат;
- по точкам в глобальной системе координат;
- по скоростям в локальной и глобальной системах координат;
- по ускорениям;
- по тяге и ориентации/угловым скоростям аппарата.

Алгоритмы должны быть или уникальными или иметь принципиальные отличия от общедоступных.

1.3.1. Управляющие воздействия

Программное управление автопилотом извне выполняется с помощью трех MAVLink-сообщений:

- [SET_POSITION_TARGET_LOCAL_NED](#);
- [SET_POSITION_TARGET_GLOBAL_INT](#);

- [SET_ATTITUDE_TARGET](#).

Сообщения ROS/MAVROS, соответствующие MAVLink-сообщениям:

- [PositionTarget](#);
- [GlobalPositionTarget](#);
- [AttitudeTarget](#).

Код обработки на стороне автопилота (для справки и уточнения деталей реализации):

- [SET_POSITION_TARGET_LOCAL_NED](#);
- [SET_POSITION_TARGET_GLOBAL_INT](#);
- [SET_ATTITUDE_TARGET](#).

Первое сообщение позволяет управлять в локальной системе координат по точкам, скоростям, ускорениям, рысканию и скорости рыскания.

Второе — в глобальной системе координат по точкам (широта, долгота, высота), скоростям, ускорениям, рысканию и скорости рыскания.

Третье — по положению в пространстве (кватерниону), скоростям крена, тангажа и рыскания, нормированному значению тяги.

В каждом сообщении возможны разные комбинации управления, которые задаются специальным полем-маской. Не все комбинации могут приниматься автопилотом, для уточнения необходимо обращаться к исходным кодам (см. выше). Примеры реализации управления по точкам и скоростям в локальной системе координат представлены в файлах:

- ros1-group — пример управления группой аппаратов через ROS;
- ros2 — пример для студенческой гонки;
- ros2-group — пример управления группой аппаратов через ROS 2;
- ros2-race-final-example — пример для студенческой гонки для финала.

1.3.2. Изменение коэффициентов ПИД-регуляторов автопилота

Значения коэффициентов ПИД-регуляторов автопилота выведены в соответствующие параметры (описание параметров см. в разделе [Модели аппаратов](#)). Описание ПИД-регуляторов и рекомендации по их изменению представлены в Multicopter PID Tuning Guide [Basic](#) и [Advanced](#).

1.4. Формации

Формации представляют собой совокупность точек пространства, в каждой из которых должен находиться один (любой) из аппаратов группы.

1.5. Использование камеры аппарата



При включении камеры происходит повышение нагрузки на CPU и GPU. Получаемый FPS зависит от технических характеристик ПК. Чтобы повысить FPS (в рамках ограничения), можно уменьшить размер окна Симулятора.



Чтобы включить камеру в окне Симулятора, нажмите на кнопку

Для получения изображений с камер аппарата используйте [ROS 2](#). Для получения изображений рекомендуется [написать подписчика](#) с использованием библиотеки [image_transport](#).

1.6. Горячие клавиши переключения видов

В Симуляторе доступны следующие виды:

- вид от первого лица (F1);
- вид от третьего лица (F2);
- вид со свободной камеры (F3);
- вид от лица пилота (F4);
- вид от третьего лица для группы моделей аппаратов (F5);
- вид с автоматически управляемой камеры (только в дисциплине «Синхронный полёт»; переключение по F6).

По клавише С происходит циклическое переключение между видами.

Переключение между аппаратами осуществляется:

- по номеру аппарата (нумерация начинается с 1) с помощью цифровых клавиш (если при этом нажать клавишу Shift, к номеру добавляется 10);
- на следующий и предыдущий аппараты с помощью клавиш X и Z соответственно.

1.7. Переключение на Vulkan

По умолчанию Симулятор использует OpenGL, но для повышения производительности имеется возможность переключиться на Vulkan.



Стабильная работа при переключении с OpenGL на Vulkan не гарантируется.

Для переключения на Vulkan в файле client/start.sh после первой строки добавьте строку:

```
GAPI="vk"
```

2. Начало работы

Запустите кластер симулятора:

```
./cluster.sh settings/ИМЯ.json
```

ИМЯ — имя файла параметров запуска, которое зависит от дисциплины и дивизиона (один из json-файлов директории settings).



Устанавливая и запуская Симулятор, вы соглашаетесь с условиями лицензии.

3. Дисциплины

Интерфейс продукта позволяет участвовать в следующих дисциплинах:

- [Синхронный полёт](#);
- [Командная гонка](#);
- [Уход от столкновения](#).

3.1. Синхронный полёт

В рамках дисциплины «Синхронный полет» задача команды — разработать и запрограммировать алгоритм, позволяющий группе из 12 беспилотных аппаратов преодолеть гоночную трассу, выстраиваясь перед трибунами зрителей в геометрические формации различной сложности.

3.1.1. Гоночная трасса

Гоночная трасса представляет собой футбольный стадион.



Рис. 1. Синхронный полёт. Трасса

Полет группы аппаратов осуществляется над выделенными зонами поля по направлению против часовой стрелки.

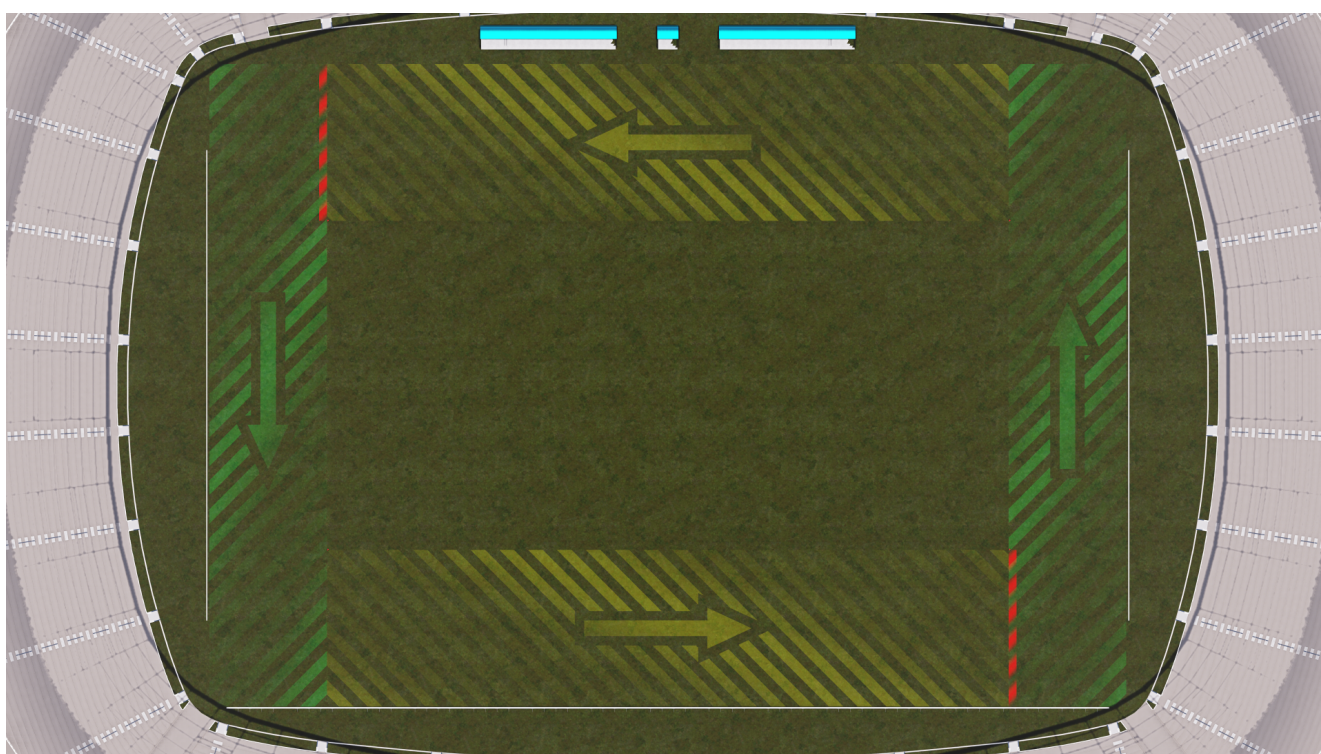


Рис. 2. Зоны поля

Зоны пролета — части поля, отмеченные желтым цветом.

Над зонами пролета группа аппаратов должна двигаться, поддерживая формацию (взаимное пространственное расположение) в соответствии с полученным заданием.

Размеры зон пролета, в метрах:

- длина — 87;
- ширина — 20;
- высота — 40.

Зоны перестроения — части поля, отмеченные зеленым цветом.

Пролетая над зонами перестроения, группа аппаратов должна перестроиться в новую формацию в соответствии с полученным заданием.

Размеры зон перестроения в метрах:

- длина — 82;
- ширина — 15;
- высота — 40.

Центры зон пролета расположены по следующим координатам: (0, 31) и (0, -31). Центры зон перестроения: (51, 0) и (-51, 0).

Место старта находится в первой зоне перестроения (51, 0).

3.1.2. Формации

Формации задаются в относительной системе координат (базис системы совпадает с базисом системы симулятора, а точка отсчета произвольна).

Например, построение в виде буквы «Т» из шести аппаратов может быть задано в таком виде:

$(0, -3, 11), (0, -1, 11), (0, 1, 11), (0, 3, 11), (0, 0, 8), (0, 0, 5)$

3.1.3. Задание

Задача команды — разработать алгоритм управления группой аппаратов, который автоматически:

- распределит месторасположение аппаратов в каждой из формаций;
- выдаст управляющие воздействия на аппараты для поддержания формаций;
- выдаст управляющие воздействия на аппараты для перестроения между формациями;
- выдаст управляющие воздействия на аппараты для перемещения по гоночной трассе для выполнения полного полетного задания.

Для Студенческого дивизиона

Полное полетное задание в виде последовательности и набора необходимых для выполнения формаций известно заранее.

- Крест:

$[[1.5, 0, 8], [-1.5, 0, 8], [-4.5, 0, 8], [4.5, 0, 8], [1.5, 0, 11], [-1.5, 0, 11], [1.5, 0, 5], [-1.5, 0, 5], [-4.5, 0, 5], [4.5, 0, 5], [1.5, 0, 2], [-1.5, 0, 2]]$

- Стрелка:

$[[5, 0, 6], [-5, 0, 6], [-5, 0, 7.5], [-2.5, 0, 7.5], [0, 0, 7.5], [2.5, 0, 7.75], [0, 0, 9.5], [2.5, 0, 4.25], [0, 0, 2.5], [-5, 0, 4.5], [-2.5, 0, 4.5], [0, 0, 4.5]]$

- Шеврон:

$[[4, 0, 9], [4, 0, 3], [3, 0, 6], [2, 0, 9], [2, 0, 3], [1, 0, 6], [-1, 0, 9], [-1, 0, 3], [-2, 0, 6], [-3, 0, 9], [-3, 0, 3], [-4, 0, 6]]$

- Самолёт:

$[[0.5, 0, 5], [3, 0, 5], [0.5, 0, 8], [3, 0, 8], [-2, 0, 5], [-2, 0, 8], [-2, 0, 11], [-2, 0, 2], [-5, 0, 5], [-5, 0, 8], [5.5, 0, 6.5], [-3.5, 0, 6.5]]$

- Пакмен:

$[[0, 0, 6], [-2.83, 0, 8.83], [-2.83, 0, 3.17], [-1.42, 0, 7.42], [-1.42, 0, 4.58], [4, 0, 6], [3.33, 0, 8.22], [1.53, 0, 9.7], [-0.78, 0, 9.92], [3.33, 0, 3.78], [1.53, 0, 2.3], [-0.78, 0, 2.08]]$

- Галактика:

```
[[0.78, 0, 8.4], [0, 0, 9.5], [1, 0, 7.23], [3.46, 0, 3.5], [2.12, 0, 3.38], [0.87, 0, 6], [1, 0, 3.77], [0, 0, 4.5], [-3.46, 0, 3.5], [-2.9, 0, 4.72], [-2, 0, 5.5], [-0.87, 0, 6]]
```

В каждой зоне пролета (до выхода из нее) должна поддерживаться одна формация согласно полученному полетному заданию. После чего в зоне перестроения должно быть выполнено перестроение в следующую по полетному заданию формацию, и поддержание созданной формации в ближайшей зоне пролета по направлению движения (показано стрелками).

Для Профессионального дивизиона

При запуске Симулятора автоматически выдается полетное задание в виде необходимой для выполнения формации в ближайшей зоне пролета по направлению движения (показано стрелками). В тот момент, когда все аппараты переходят из зоны пролета в зону перестроения (красные линии), выдается следующее полетное задание в ближайшей зоне пролета по направлению движения.

Выдача полетного задания происходит с помощью системы ROS 2. При запуске симулятора также запускается ROS 2-нода, публикующая в ROS 2-топик `formation` текущие относительные координаты для построения формации в формате

```
geometry_msgs/msg/PoseArray
```

3.1.4. Запуск

Команда для запуска Симулятора выглядит следующим образом.

Студенческий дивизион:

```
./cluster.sh settings/formation_stud_final_settings.json
```

Профессиональный дивизион:

```
./cluster.sh settings/formation_prof_final_settings_n.json
```

Доступны несколько вариантов параметров, `n` — номер варианта.

3.1.5. Критерии оценивания

Измеряются следующие критерии:

- время прохождения трассы.

Таймер запускается при взлёте хотя бы одного аппарата, а завершается после прохождения зоны пролёта последней формации.

- Количество столкновений.

Повторное столкновение засчитывается только через 0,5 секунды после первого столкновения. Два столкновения учитываются единожды, если между ними прошло менее 0,5 секунды.

- Длительность вылета за пределы ограничительной зоны.

Суммарное время, проведённое как минимум одним аппаратом за пределами ограничительной зоны во время прохождения зоны полёта.

- Отклонение от формации.

При пролёте зоны показа в каждый момент времени через равные промежутки времени измеряется среднеквадратическое отклонение (СКО) положения аппаратов от точек формации. Для каждой зоны показа вычисляется среднеквадратическое значение СКО (СКСКО). После этого вычисляется сумма СКСКО по зонам полёта.

Время зачётной попытки вычисляется на основе времени прохождения трассы со штрафами по формуле:

Время зачетной попытки = Время прохождения трассы + A_f * Количество столкновений + B_f * Длительность вылета за пределы ограничительной зоны + C_f * Отклонение от формации

Коэффициенты имеют следующие значения:

- $A_f = 5$ с;
- $B_f = 15$;
- $C_f = 200$ с/м.

По окончании гонки выводится сообщение с информацией о зачетном времени с учётом штрафного:

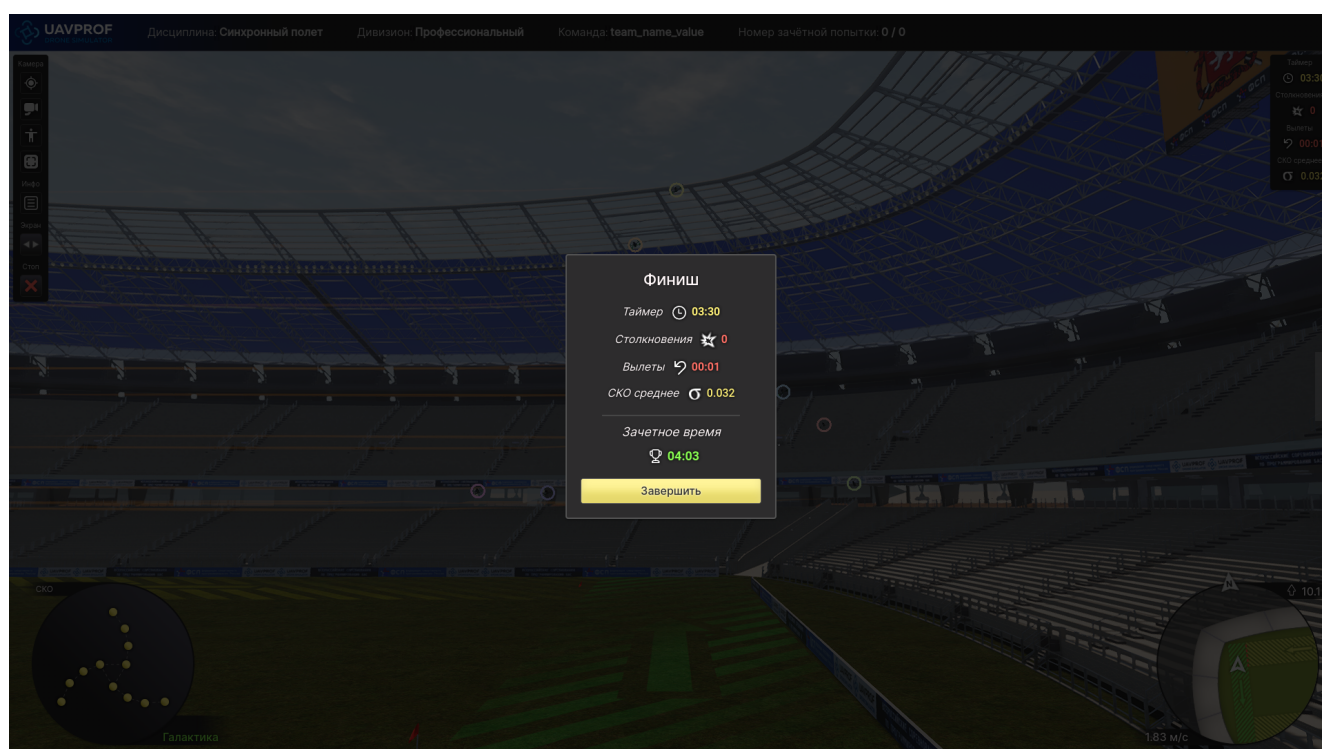


Рис. 3. Финиш

3.2. Командная гонка

В рамках дисциплины «Командная гонка» задача команды — разработать и продемонстрировать работоспособность алгоритма, позволяющего беспилотному аппарату

пролететь гоночную трассу с использованием технического зрения.

3.2.1. Гоночная трасса

Гоночная трасса представляет собой последовательность из 7 ворот в помещении с размерами 25.94 x 39.19 x 8.41 м. Трасса является круговой: одни ворота являются стартовыми/финишными, а остальные — промежуточными. Последовательность ворот на трассе может изменяться. Трасса находится в зоне размером 20 x 25 x 6 м.

Аппарату необходимо взлететь со взлётной площадки и пролететь 3 круга.

Ворота — объект с квадратным отверстием 1.54 x 1.54 м, через которое необходимо пролететь аппарату. Чтобы пролететь ворота, аппарату необходимо центром масс пересечь плоскость отверстия (внутри отверстия) в правильном направлении.

На воротах со стороны, в которую нужно влететь, рядом с углами отверстия нанесены ArUco-маркеры размером 0.19 x 0.19 м (коды маркеров взяты из [словаря DICT_4X4_250 библиотеки OpenCV](#)). Номер маркера в словаре вычисляется по формуле:

$$M = N * 4 + C$$

Параметры формулы:

- $N \in [0..6]$ — последовательный номер ворот;
- $C \in [0..3]$ — номер угла:
 - 0 — левый нижний;
 - 1 — правый нижний;
 - 2 — правый верхний;
 - 3 — левый верхний.

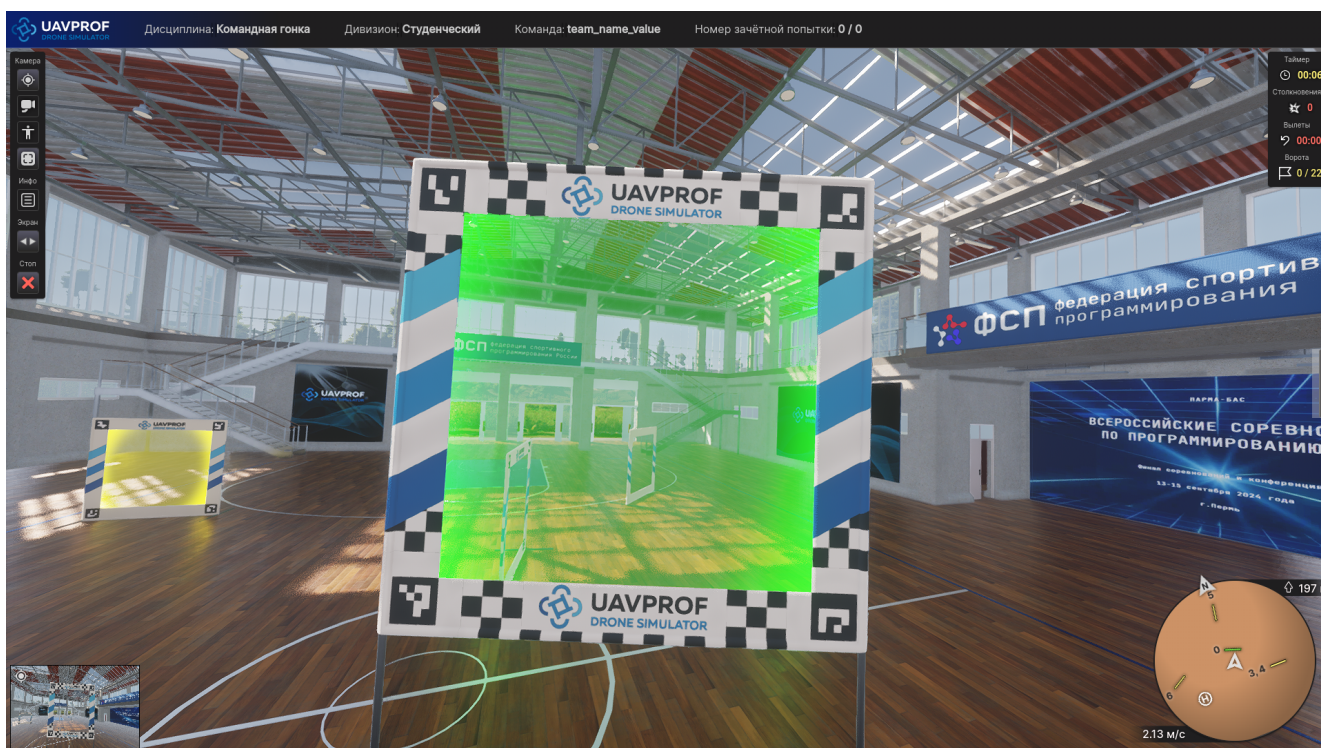


Рис. 4. Внешний вид ворот

Для визуализации (не присутствуют на изображениях с камеры аппарата) показаны: зона трассы (видна вблизи), линия трассы и отмечены ворота. Ворота, которые необходимо пролететь, окрашены в зелёный цвет. При нарушении направления пролёта ворота окрашиваются в красный цвет. Следующие ворота — жёлтые, предыдущие — красные.

Все объекты трассы являются непроходимой преградой для аппаратов — на них работает механизм упругого столкновения. В большинстве случаев это правило действует и для помещения (вне зоны трассы могут встречаться невидимые преграды).

3.2.2. Аппарат

Выберите модель аппарата в зависимости от дивизиона:

- для Студенческого дивизиона — FPV F5D Auton LPS;
- для Профессионального дивизиона — FPV F5D Auton.

3.2.3. Задание

Задача аппарата — взлететь и пролететь 3 круга гоночной трассы.

Задача команды — разработать алгоритм управления аппаратом, который автоматически выдаст управляющие воздействия на аппарат для выполнения задачи аппарата.

3.2.4. Запуск

Команда для запуска Симулятора выглядит следующим образом.

Студенческий дивизион:

```
./cluster.sh settings/race_stud_final_settings_n.json
```

Доступны несколько вариантов параметров, n — номер варианта.

Профессиональный дивизион:

```
./cluster.sh settings/race_prof_final_settings_n.json
```

Доступны несколько вариантов параметров, n — номер варианта.

3.2.5. Критерии оценивания

Измеряются следующие критерии:

- время прохождения трассы.

Таймер запускается при взлёте аппарата, а завершается после прохождения последних ворот.

- Количество столкновений.

Повторное столкновение засчитывается только через 0,5 секунды после первого столкновения. Два столкновения учитываются единожды, если между ними прошло менее 0,5 секунды.

- Длительность вылета за пределы ограничительной зоны.

Суммарное время, проведённое аппаратом за пределами ограничительной зоны.

Время зачётной попытки вычисляется на основе времени прохождения трассы со штрафами по формуле:

Время зачётной попытки = Время прохождения трассы + A_r * Количество столкновений + B_r * Длительность вылета за пределы ограничительной зоны

Коэффициенты имеют следующие значения:

- $A_r = 15$ с;
- $B_r = 20$.

3.3. Уход от столкновения

В рамках дисциплины «Уход от столкновения» задача команды — реализовать алгоритм управления, позволяющий БПЛА пролететь заданный маршрут, избежав столкновений с препятствиями (воздушным шаром / птицами).

3.3.1. Гоночная трасса

Полигон представляет собой участок горной пересеченной местности.



Рис. 5. Уход от столкновения. Трасса

Маршрут представляет собой ломаную линию, заданную координатами ее вершин (точек маршрута). Полет по маршруту начинается при взлете аппарата и заканчивается при его посадке на посадочную площадку с расстоянием 3 метра до центра этой площадки. При полете по маршруту требуется строгое поддержание траектории. Допустимая зона находится вокруг

линии маршрута не дальше некоторого расстояния R от нее. Значение R в отборочном онлайн туре равно 7.5 метров.

В качестве препятствий выступают воздушный шар / птицы, местоположение которых может изменяться в процессе полета маршрута.

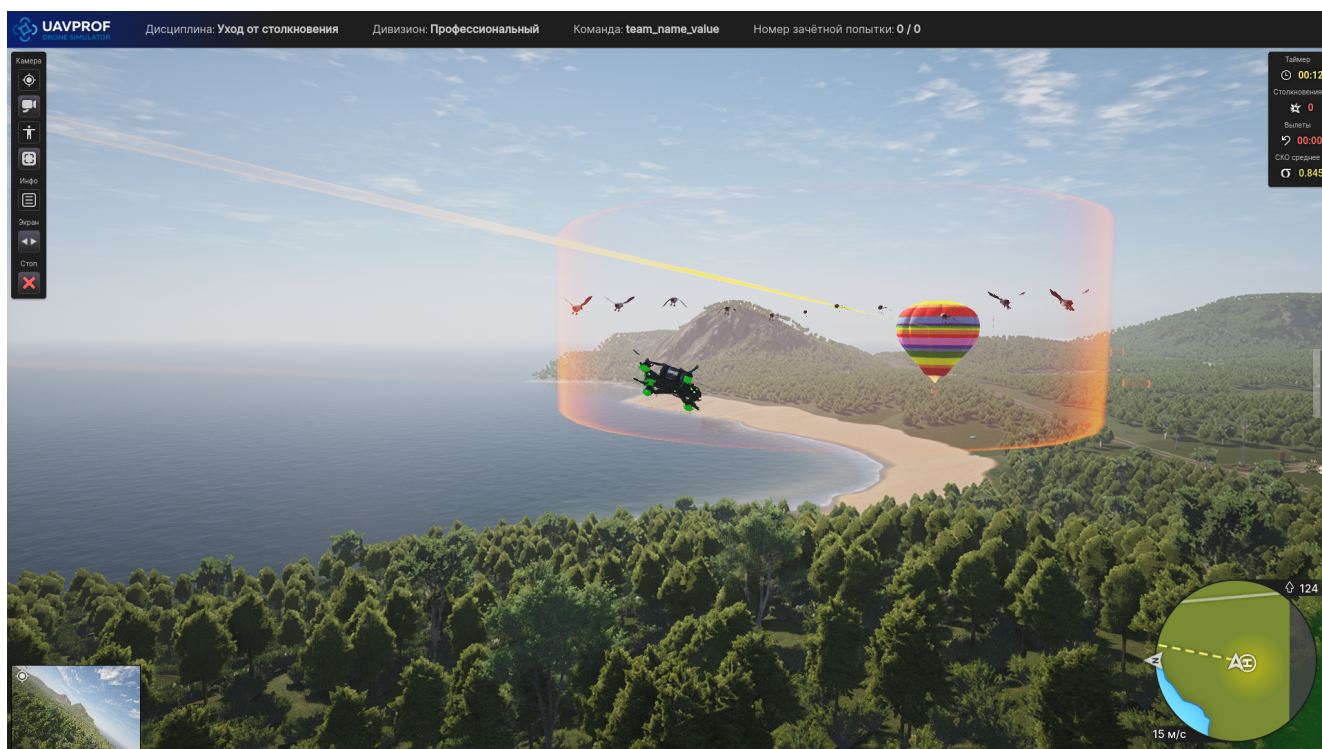


Рис. 6. Пример препятствия

Птицы имеют внешний коллайдер (объект, определяющий форму и размер зоны столкновения). Птиц следует облетать:

- на расстоянии 2-3 метра выше их траектории полета;
- 3-4 метра ниже их траектории;
- 5 метров слева/справа.

Воздушный шар обладает коллайдером, соответствующим его визуальной модели; его целесообразно облетать слева или справа.

3.3.2. Задание

Задача команды — разработать алгоритм управления аппаратом, который автоматически:

- выдаст управляющее воздействие на аппарат для перемещения по маршруту;
- определит препятствия, с которыми может произойти столкновение;
- выдаст управляющее воздействие для избегания столкновений.

Для Студенческого дивизиона положение препятствий, а также их габаритные размеры выдаются в течение полета аппарата, при нахождении препятствия на расстоянии обнаружения L от аппарата. В отборочном онлайн туре значение L будет статическим и равным 10 метрам.

Для Профессионального дивизиона параметры динамических препятствий необходимо определять с помощью оптических камер аппарата.

Координаты точек маршрута в локальной системе координат Симулятора:

```
[ 549.75,    207.8,    98.56658],
[ 549.75,    207.8,   125.0    ],
[ 580.0,     366.0,   125.0    ],
[ 624.55501, 598.40931, 70.0    ],
[ 928.84280, 947.73771, 95.0    ],
[ 965.0,     1238.0,  100.0    ],
[1108.28750, 1443.92409, 105.0   ],
[ 933.8,     1701.4,  140.0    ],
[ 933.8,     1701.4,  107.66052]
```

Допустимое расстояние **R** вокруг линии маршрута, в метрах: 7.5

Расстояние обнаружения препятствий (для Студенческого дивизиона) **L**, в метрах: 10

Выдача информации о препятствиях (для Студенческого дивизиона) происходит с помощью системы ROS 2. При запуске симулятора также запускается ROS 2-нода, публикующая в ROS 2-топик `obstacles` информацию о препятствии в формате строки

```
std_msg/msg/String
```

Формат строки, публикуемой в топик; значения разделены пробелами:

```
obj_name x y z scale_x scale_y scale_z
```

Значения в строке:

- **obj_name** — имя препятствия;
- **x y z** — координаты препятствия;
- **scale_x scale_y scale_z** — габариты препятствия (в метрах).

В рамках отборочного онлайн-тура **obj_name** может принимать значения:

- **hot_air_balloon** — воздушный шар;
- **eagles** и **ducks** — группы птиц.

3.3.3. Запуск

Команда для запуска Симулятора выглядит следующим образом.

Студенческий дивизион:

```
./cluster.sh settings/avoid_stud_final_settings_n.json
```

Доступны несколько вариантов параметров, **n** — номер варианта.

Профессиональный дивизион:

```
./cluster.sh settings/avoid_prof_final_settings_n.json
```

Доступны несколько вариантов параметров, n — номер варианта.

3.3.4. Критерии оценивания

Оценивается время, затраченное на прохождение задания. При этом из трех зачетных попыток выбирается попытка с наименьшим временем прохождения. Отсчет времени начинается в момент пересечения аппаратом границы зоны старта и заканчивается в момент пересечения аппаратом границы зоны посадки.

Измеряются следующие критерии:

- время прохождения трассы.

Таймер запускается при взлёте аппарата, а завершается при приземлении аппарата на посадочную площадку.

- Количество столкновений.

Повторное столкновение засчитывается только через 0,5 секунды после первого столкновения. Два столкновения учитываются единожды, если между ними прошло менее 0,5 секунды.

- Длительность вылета за пределы ограничительной зоны.

Суммарное время, проведённое аппаратом за пределами ограничительной зоны.

- Отклонение от маршрута.

При пролёте маршрута в каждый момент времени через равные промежутки времени измеряется отклонение аппарата от ближайшей точки на ломаной маршрута. После преодоления маршрута вычисляется среднеквадратическое отклонение (СКО) по всем измеренным отклонениям.

Время зачётной попытки вычисляется на основе времени прохождения трассы со штрафами по формуле:

Время зачетной попытки = Время прохождения трассы + A_a * Количество столкновений + B_a * Длительность вылета за пределы ограничительной зоны + C_a * Отклонение от маршрута

Коэффициенты имеют следующие значения:

- $A_a = 50$ с;
- $B_a = 100$;
- $C_a = 100$ с/м.

4. Судейство

Для проведения судейства в поставке имеются файлы вариантов параметров для запуска Симулятора, которые недоступны участникам.

Синхронный полёт

- Студенческий дивизион: `formation_stud_judge_settings.json`
- Профессиональный дивизион: `formation_prof_judge_settings.json`

Командная гонка

- Студенческий дивизион: `race_stud_judge_settings.json`
- Профессиональный дивизион: `race_prof_judge_settings.json`

Уход от столкновений

- Студенческий дивизион: `avoid_stud_judge_settings.json`
- Профессиональный дивизион: `avoid_prof_judge_settings.json`

История изменений

15.11.2024

Документ создан.