Цель и задачи работы

Название **Реализация фигуры высшего пилотажа на lua скрипте под Ardupilot**

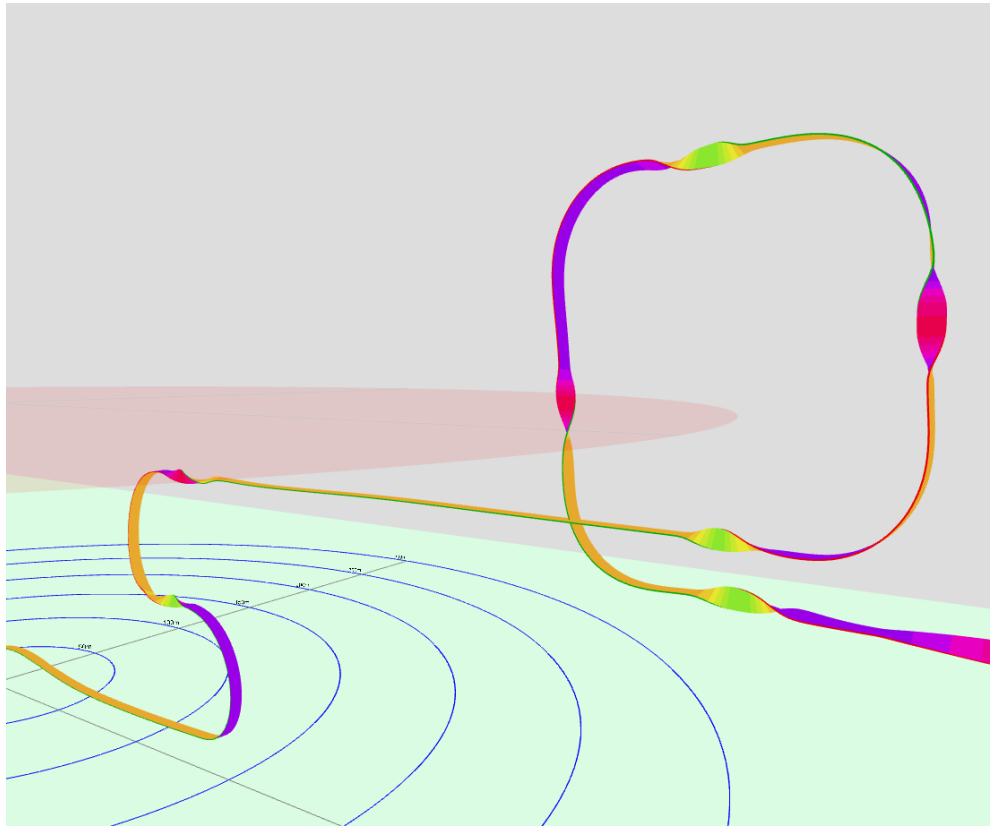
Цель Определить способность самолета с прошивкой Ardupilot выполнить нужную акробатичскую фигуру высшего пилотажа

Задачи

1. Создание скрипта реализации фигуры пилотажа
2. Создание полетного задания
3. Моделирование запуска полетного задания в SITL

Методы решения задач

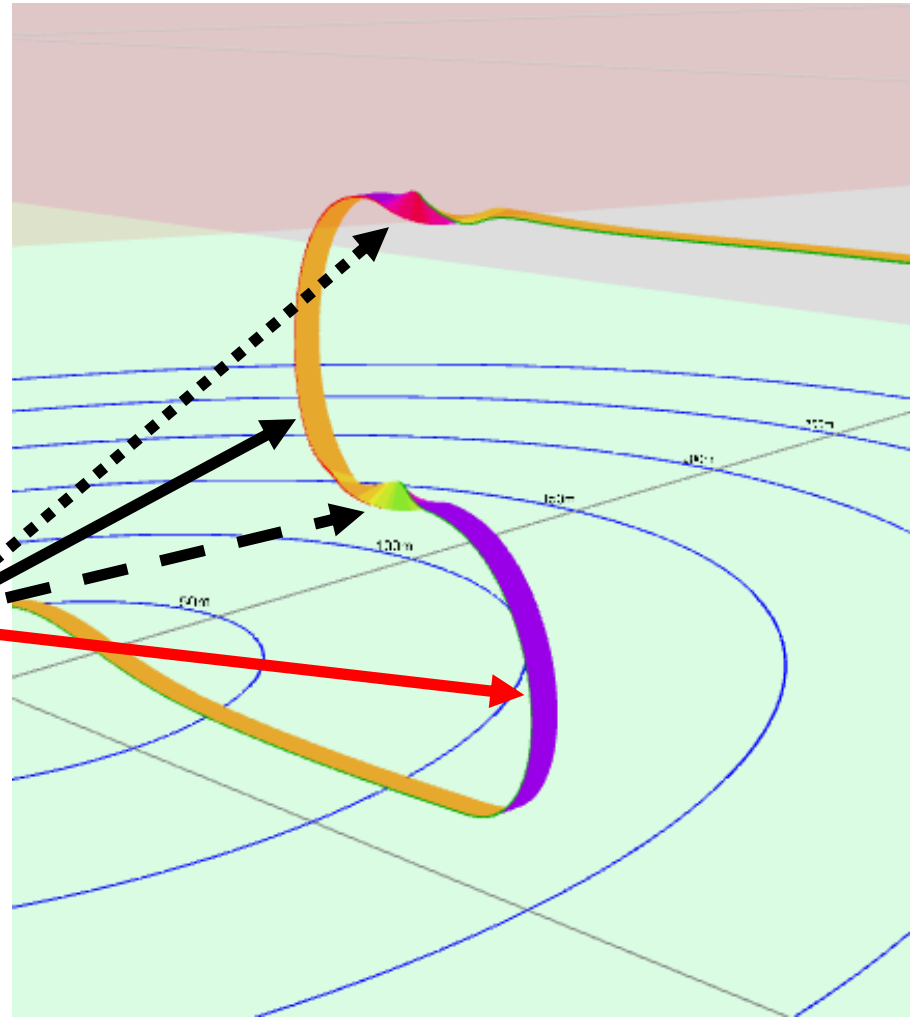
- Фигура высшего пилотажа реализуется с помощью библиотеки plane_aerobatics.lua
- Выбрана реализация фигур под условными названиями S петля Иммельмана и квадратная петля.
- SITL осуществляется с помощью Mission Planner с загрузкой plane simulation (модель plane-3d)
- Визуальный анализ качества выполнения полетного задания осуществляется с помощью сервиса https://www.flightcoach.org/ribbon/plotter.html



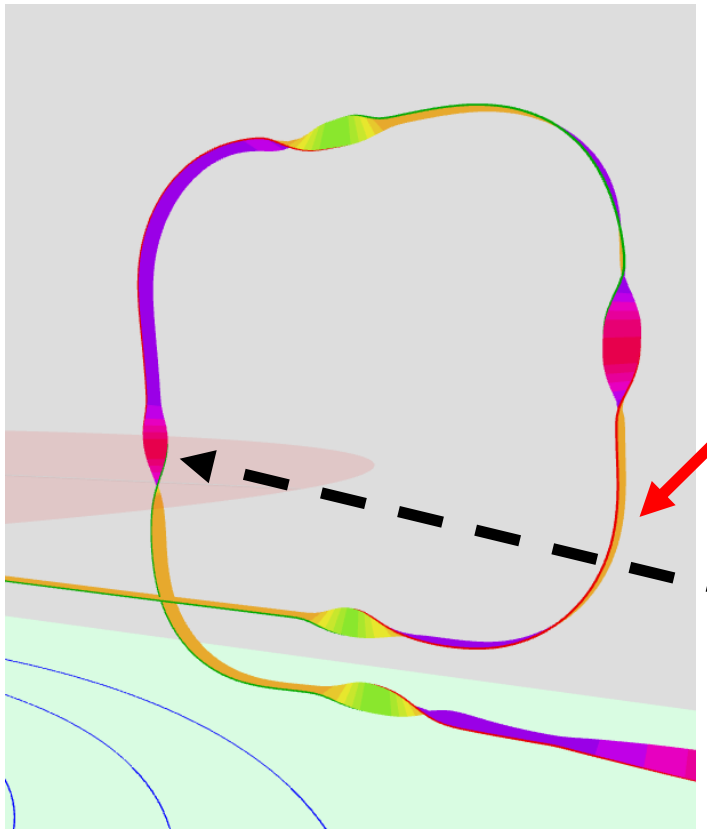
Реализация фигуры «S петля Иммельмана»

Библиотека plane_aerobatics.lua разделяет сложные фигуры пилотажа на простые элементы (движение по дуге, движение прямо, переворот и т.д.). Элементы могут выполняться последовательно и/или одновременно.

```
function immelmann_s_turn(r, arg2, arg3, arg4)
local rabs = math.abs(r)
return make_paths("immelmann_s_turn", {
  { path_vertical_arc(r, 180),     roll_angle(0) },
  { path_straight(rabs),           roll_angle(180) },
  { path_vertical_arc(-r, 180),    roll_angle(0) },
  { path_straight(rabs),           roll_angle(180) },
  })
end
```



Реализация фигуры «Квадратная петля»



```
function square_loop(radius, height, arg3, arg4)   -- square
  local h = height - (2 * radius)
  local side = h / math.cos(math.rad(45))
  local base = (2 * (h + radius)) - 2 * radius
  return make_paths("square_loop", {
        { path_straight(base * 1/5),          roll_angle(180) },
        { path_straight(base * 2/5),          roll_angle(0) },
        { path_vertical_arc(radius, 90) ,       roll_angle(0) },
        { path_straight(side*2/9),          roll_angle(0) },
        { path_straight(side*2/9),          roll_angle(90) },
        { path_straight(side*1/9),          roll_angle(0) },
        { path_straight(side*2/9),          roll_angle(90) },
        { path_straight(side*2/9),          roll_angle(0) },
        { path_vertical_arc(radius, 90),        roll_angle(0) },
        { path_straight(side*2/9),          roll_angle(0) },
        { path_straight(side*2/9),          roll_angle(90) },
        { path_straight(side*1/9),          roll_angle(0) },
        { path_straight(side*2/9),          roll_angle(90) },
        { path_straight(side*2/9),          roll_angle(0) },
        { path_vertical_arc(radius, 90),        roll_angle(0) },
        { path_straight(base * 2/5),          roll_angle(0) },
        { path_straight(base * 1/5),          roll_angle(180) },
        { path_straight(base * 2/5),          roll_angle(0) },
        { path_vertical_arc(radius, 90),        roll_angle(0) },
        { path_straight(side*2/9),          roll_angle(0) },
        { path_straight(side*2/9),          roll_angle(90) },
        { path_straight(side*1/9),          roll_angle(0) },
        { path_straight(side*2/9),          roll_angle(90) },
        { path_straight(base*2/5),          roll_angle(0) },
    })
end
```