

Московский государственный технический университет
имени Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Системы автоматического управления»

А.А. Бушев, А.И. Жильцов, М.В. Фетисов

**Методические указания
по выполнению лабораторных работ по дисциплине
«Управляющие комплексы беспилотных летательных ап-
паратов»**

Электронное учебное издание

УДК 004.942+ 519.85+519.876.5

Рецензент:

А.А. Бушев, А.И. Жильцов, М.В. Фетисов.

Методические указания по выполнению лабораторных работ по дисциплине «Управляющие комплексы беспилотных летательных аппаратов». Электронное учебное издание. – М.: МГТУ имени Н.Э. Баумана, 2022, 41 с.

Учебное издание содержит необходимые базовые теоретические вводные для решения задачи математического моделирования и построения грубых систем управления беспилотным летательным аппаратом, а также указания и требования к порядку выполнения лабораторных работ. Определяются цели и объем, требования к отчетам, а также приводятся варианты заданий и примерный список контрольных вопросов для защиты выполненных заданий. Приводятся примеры выполнения лабораторных работ с использованием системы моделирования SIMODO stars. Также даётся краткое описание этой системы моделирования.

Для студентов первого курса магистратуры ИУ1 МГТУ имени Н.Э. Баумана специальности 24.05.06 «Системы управления летательными аппаратами».

Рекомендовано учебно-методической комиссией факультета «Информатика и системы управления» МГТУ им. Н.Э. Баумана

Электронное учебное издание

А.А. Бушев, А.И. Жильцов, М.В. Фетисов

**Методические указания по выполнению лабораторных работ и домашнего задания по дисциплине
«Управляющие комплексы беспилотных летательных аппаратов»**

Оглавление

ВВЕДЕНИЕ	5
1 АЛГОРИТМЫ И СТРУКТУРЫ СИСТЕМ СТАБИЛИЗАЦИИ БЛА	6
1.1 Поддержание заданной высоты полёта	12
1.2 Поддержание заданной скорости полёта	14
1.3 Поддержание заданной боковой перегрузки	16
1.4 Поддержание заданного курса полёта	17
1.5 Прохождение через окрестности заданной точки	18
1.6 Прохождение по маршруту	19
2 АЛГОРИТМЫ И СТРУКТУРЫ СИСТЕМЫ УПРАВЛЕНИЯ ПОЛЕТОМ БЛА	23
2.1 Автоматический взлёт	23
2.2 Автоматическая посадка с выравниванием	24
2.3 Посадка в заданную точку	24
2.4 Взлёт, прохождение ППМ и посадка	25
2.5 Автоматическое уклонение от потенциальной угрозы	26
2.6 Сброс объекта на заданный участок	27
2.7 Моделирование группового взаимодействия	28
ПРИЛОЖЕНИЕ А. КРАТКОЕ ОПИСАНИЕ SIMODO STARS	31
А.1 Общее описание	31
А.2 Интегрированная среда разработки SIMODO STARS	31
А.3 Язык описания алгоритмов SCRIPTCO	34
А.4 Язык описания дифференциальных уравнений DIFF	35
А.5 Встроенные модули	36
ЛИТЕРАТУРА	40
ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СОКРАЩЕНИЙ И ТЕРМИНОВ	41

Аннотация

Учебное издание содержит необходимые базовые теоретические вводные для решения задачи математического моделирования и построения грубых систем управления беспилотным летательным аппаратом, а также указания и требования к порядку выполнения лабораторных по дисциплине «Управляющие комплексы беспилотных летательных аппаратов». Определяются цели и объем, требования к отчетам, а также приводятся варианты заданий и примерный перечень контрольных вопросов для защиты выполненных заданий. Приводятся примеры выполнения лабораторных работ с использованием системы моделирования SIMO-DO stars. Также даётся краткое описание этой системы моделирования.

Для студентов первого курса магистратуры ИУ1 МГТУ имени Н.Э. Баумана специальности 24.05.06 «Системы управления летательными аппаратами».

Введение

Приведённый лабораторный практикум направлен на решение нескольких задач.

Во-первых, демонстрируется универсальный подход к разработке структур и алгоритмов систем управления движущимися объектами.

Во-вторых, предлагается физическое обоснование этой универсальности.

В-третьих, осуществляется (выполняется) математическое моделирование динамических процессов, характеризующих управляемое движение, при помощи перспективной системы моделирования SIMODO stars. Для сравнения можно выполнить математическое моделирование при помощи широко известной системы MATLAB.

Универсальность подхода к управлению движущимся объектом заключается в обеспечении сбалансированных движений по каждому каналу управления.

Пространственное движение рассматривается при помощи математической модели, описывающей управляемое движение центра масс летательного аппарата (материальной точки).

Порядок выполнения лабораторных работ описывается с использованием системы моделирования SIMODO stars. Краткое описание системы моделирования SIMODO stars приведено в Приложение А.

Также для успешного выполнения работ практикума необходимо самостоятельное использование учебной и справочной литературы и других источников.

Практикум по дисциплине «Управляющие комплексы беспилотных летательных аппаратов» обеспечивает базовые знания для формирования профессиональных компетенций, обозначенных в соответствии с самостоятельно установленным образовательным стандартом, основной профессиональной образовательной программой по направлению подготовки 24.05.06 «Системы управления летательными аппаратами»:

- способностью понимать сущность и значение научно-технической информации и соблюдения требований информационной безопасности (СОПК-1);
- анализировать основные этапы и закономерности развития общества (СОК-2);
- работать в команде (СОК-6);
- собирать и анализировать научно-техническую информацию (СПК-1);
- самостоятельно выполнять лабораторные и натурные исследования и проводить эксперименты (СПК-2);
- способностью разрабатывать программу и план натурных испытаний (СПСК-29.9).

1 Алгоритмы и структуры систем стабилизации БЛА

Уравнение движения летательного аппарата можно описать дифференциальным уравнением в векторной форме (1).

$$\dot{X} = f(X, U), \quad (1)$$

где X – вектор векторов X_1, X_2, X_3 и X_4 , которые, соответственно, описывают кинематику и динамику движений центра масс и вокруг центра масс, а U – управляющие воздействия на движение центра масс и вокруг центра масс, соответственно (2).

$$\begin{cases} \dot{X}_1 = f_1(X_1, X_2) & \text{– кинематика движения центра масс;} \\ \dot{X}_2 = f_2(X_2, U_1) & \text{– динамика движения центра масс;} \\ \dot{X}_3 = f_3(X_3, X_4) & \text{– кинематика движения вокруг центра масс;} \\ \dot{X}_4 = f_4(X_4, U_2) & \text{– динамика движения вокруг центра масс.} \end{cases} \quad (2)$$

Теперь раскроем эти вектора. Вектор координат центра масс X_1 представлен в нормальной земной системе координат (СК) (3).

$$X_1 = \begin{pmatrix} x_g \\ y_g \\ z_g \end{pmatrix}. \quad (3)$$

Вектор скоростей центра масс в нормальной земной СК (4).

$$X_2 = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}. \quad (4)$$

Вектор углового положения будет рассматриваться как совокупность углов рыскания, тангажа и крена (5).

$$X_3 = \begin{pmatrix} \psi \\ \vartheta \\ \gamma \end{pmatrix}. \quad (5)$$

Вектор угловых скоростей раскрывается в формуле (6).

$$X_4 = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}. \quad (6)$$

Вектор перегрузок, действующих на центр масс раскрыт в формуле (7).

$$U_1 = \begin{pmatrix} n_{x_t} \\ n_{y_t} \\ n_{z_t} \end{pmatrix}. \quad (7)$$

Вектор управления раскрывается в отклонения элеронов, рулей направления и высоты, а также рычага управления двигателем, соответственно (8).

$$U_2 = \begin{pmatrix} \delta_{\text{э}} \\ \delta_{\text{н}} \\ \delta_{\text{в}} \\ \delta_{\text{руд}} \end{pmatrix} \quad (8)$$

Система уравнений (9), записанная в скалярной форме, описывает движение центра масс летательного аппарата (ЛА) в скоростной СК:

$$\begin{cases} \dot{x}_g = V \cos \theta \cos \psi \\ \dot{y}_g = V \sin \theta \\ \dot{z}_g = -V \cos \theta \sin \psi \\ \dot{V} = g(n_{x_a} - \sin \theta) \\ \dot{\theta} = \frac{g}{V}(n_{y_a} \cos \gamma - \cos \theta) \\ \dot{\psi} = -\frac{g}{V \cos \theta} n_{z_a} \sin \gamma \end{cases}, \quad (9)$$

где:

V – модуль скорости центра масс в нормальной земной СК, т.е. $V = \sqrt{v_x^2 + v_y^2 + v_z^2}$;

θ – угол наклона траектории;

g – ускорение свободного падения;

ψ – угол курса.

Система уравнений (10), записанная в скалярной форме, описывает движение центра масс ЛА в траекторной СК.

$$\begin{cases} \dot{x}_g = V \cos \theta \cos \psi \\ \dot{y}_g = V \sin \theta \\ \dot{z}_g = -V \cos \theta \sin \psi \\ \dot{V} = g(n_{x_t} - \sin \theta) \\ \dot{\theta} = \frac{g}{V}(n_{y_t} - \cos \theta) \\ \dot{\psi} = -\frac{g}{V \cos \theta} n_{z_t} \end{cases} \quad (10)$$

Приведение кинематики:

$$y_1 = x_g; y_2 = y_g; y_3 = z_g \quad (11)$$

И, наконец, всё вместе:

$$\begin{cases} \dot{y}_1 = y_4 \\ \dot{y}_2 = y_5 \\ \dot{y}_3 = y_6 \\ \dot{y}_4 = \frac{gy_4}{\sqrt{y_4^2 + y_5^2 + y_6^2}} \left(n_{x_t} - \frac{y_5}{\sqrt{y_4^2 + y_6^2}} n_{y_t} \right) - \frac{gy_6}{\sqrt{y_4^2 + y_6^2}} n_{z_t} \\ \dot{y}_5 = \frac{gy_5}{\sqrt{y_4^2 + y_5^2 + y_6^2}} n_{x_t} + \frac{g\sqrt{y_4^2 + y_6^2}}{\sqrt{y_4^2 + y_5^2 + y_6^2}} n_{y_t} - g \\ \dot{y}_6 = \frac{gy_6}{\sqrt{y_4^2 + y_5^2 + y_6^2}} \left(n_{x_t} - \frac{y_5}{\sqrt{y_4^2 + y_6^2}} n_{y_t} \right) + \frac{gy_4}{\sqrt{y_4^2 + y_6^2}} n_{z_t} \end{cases} \quad (12)$$

Управление траекторией движения ЛА (12) осуществляется с помощью управляющих воздействий n_{x_t} , n_{y_t} и n_{z_t} . В зависимости от конкретной задачи управления ЛА эти управляющие воздействия вычисляются по-разному.

Рассматриваются два канала управления: управляемое движение в горизонтальной плоскости и управляемое движение в вертикальной плоскости.

Сбалансированным движением в горизонтальной плоскости является координированный разворот, выполняемый на заданной высоте. Поворот вектора скорости V_r в координированном развороте осуществляется с угловой скоростью Ω . Центроостремительное ускорение равно инерциальному центробежному $A_{цс} = \Omega V_r$. Угол курса ψ представляет собой (13).

$$\psi = \int_0^{\tau} \Omega dt + \psi(0) \quad (13)$$

На рисунке 1 представлена схема поддержки заданного углового курса.

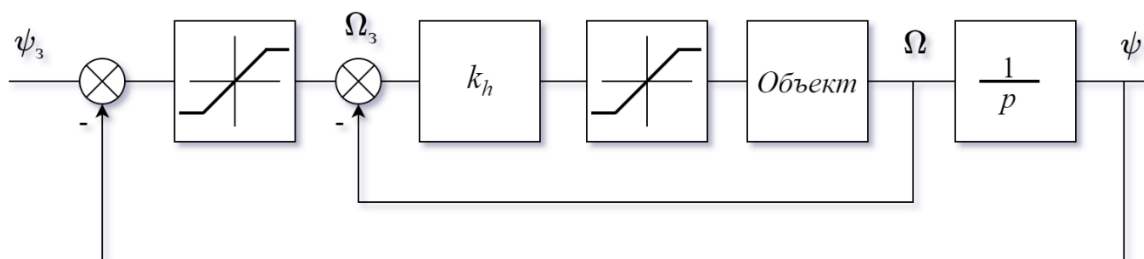


Рисунок 1. Схема поддержания заданного угла курса

Сбалансированным движением в вертикальной плоскости является движение с постоянной вертикальной скоростью. В соответствии с (12): $y_5 = const$. Высота H представляет собой (14).

$$H = y_2 = \int_0^{\tau} y_5 dt + y_2 \psi(0) \quad (14)$$

На рисунке 2 представлена схема поддержания заданной высоты.

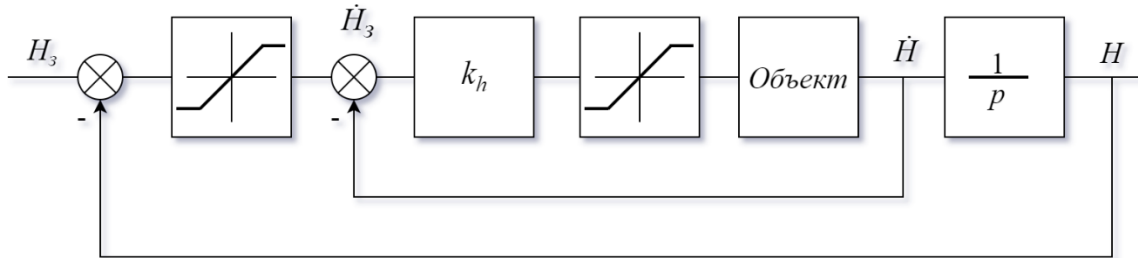


Рисунок 2. Схема поддержания заданной высоты

Объединение двух сбалансированных движений в горизонтальной и вертикальной плоскостях даёт сбалансированное пространственное винтообразное движение.

Для реального летательного аппарата такое движение может быть выполнено как движение с постоянным креном, постоянной скоростью $V = \sqrt{y_4^2 + y_5^2 + y_6^2} = const$, постоянной вертикальной скоростью $V_v = y_5 = const$ с учётом реальных ограничений, связанных с ограничением тяги силовой установки (двигателя). Крен летательного аппарата в сбалансированном движении ограничен $\pm \frac{\pi}{2}$.

Для координированного разворота, то есть для разворота с постоянной скоростью на постоянной высоте, разным углам крена γ соответствуют разные значения перегрузок $n_y = \frac{Y}{mG}$,

где Y – подъёмная сила (см. рисунок 3).

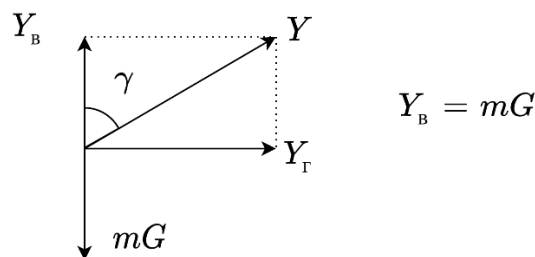


Рисунок 3. Разложение перегрузок при координированном развороте с постоянной скоростью на постоянной высоте

Для (12) $n_{yt} = \frac{Y_v}{mG}$, $n_{zt} = \frac{Y_h}{mG}$, $n_{xt} = \frac{P-X}{mG}$, где P — тяга силовой установки, X - лобовое сопротивление.

Таким образом, для системы уравнений (12), характеризующей движение центра масс летательного аппарата (ЛА), можно записать законы управления для различных режимов полёта.

Стабилизация описанных сбалансированных движений и представляет собой универсальный подход к разработке структур и алгоритмов систем управления полётом беспилотного летательного аппарата (БЛА).

Физическим обоснованием этого универсального подхода является канальная *балансировка*.

Для ЛА сбалансированное пространственное движение соответствует траектории, представляющей собой винтовую линию на вертикально ориентированном цилиндре. Этому движению удовлетворяют для математической модели (12) управляющие воздействия:

$$\begin{aligned} n_{xT} &= k_V(V_z - V); \\ \cos \theta &= \sqrt{1 - \frac{y_5^2}{y_4^2 + y_5^2 + y_6^2}}; \quad n_{yT} = \cos \theta + k_{ny}(y_{5z} - y_5); \\ n_{zT} &= \text{const}; \end{aligned} \quad (15)$$

, где θ – угол наклона траектории.

Рассматриваются различные виды формирования управляющих воздействий для моделирования движения центра масс БЛА в траекторной системе координат (12).

Для отработки траекторных задач на математической модели движения центра масс ЛА (12) достаточно формировать управляющие воздействия: n_{xT} , n_{yT} и n_{zT} .

Для реального самолёта можно также отрабатывать эти перегрузки, что в большинстве случаев система управления полётом и делает.

Но возможен и другой путь. Можно отрабатывать заданные скорости. Например, поддерживая заданную воздушную скорость V , можно одновременно отрабатывать заданную вертикальную скорость V_z и скорость Ω , характеризующую угловую скорость поворота горизонтально составляющей скорости V_z .

Такая структура, построенная как совокупность систем стабилизации внутри каждого канала управления даёт важное преимущество перед структурой, работающей на ускорениях (перегрузках).

В то время, как для математической модели движения центра масс (12) достаточно во внутренних контурах стабилизации (регулирования) использовать пропорциональное регулирование (П-регулятор), то для реального объекта с естественными ограничителями во внутренних контурах *нужно* использовать ПИ-регуляторы, что обеспечит грубость динамической системы управления.

В таком варианте организации системы управления полётом можно добиться *автоматической балансировки* летательного аппарата, или *самобалансировки*, при правильном соотношении движений центра масс и вокруг центра масс ЛА.

Это свойство оказывается очень полезным для систем управления полётом БЛА, характеристики и параметры которых точно не известны. Оказывается, что такая структура систем управления с внутренними ПИ-регуляторами позволяет сохранять управляемость объектов с повреждениями и некритическими отказами.

Система уравнений (12) удобна для физической интерпретации свойств реализуемого движения центра масс ЛА.

Кроме того, эта математическая модель естественно дополняется до полной модели, учитывающей также и движение вокруг центра масс (кинематику и динамику), в которую подключается банк аэродинамических коэффициентов.

В качестве примера новой физической интерпретации ранее известных структур, используемых в системах управления полётом, рассмотрим так называемый тангажный автомат:

$$\Delta\delta_v = k_\vartheta(\vartheta_{\text{зад}} - \vartheta) + k_\omega\omega_z \quad (16)$$

где: $\Delta\delta_v$ – отклонение руля высоты;

$\vartheta_{\text{зад}}$ – заданное значение угла тангажа;

ϑ – текущее значение угла тангажа;

ω_z – угловая скорость вращения ЛА вокруг боковой оси связанной с объектом системы координат.

k_ϑ, k_ω – коэффициенты.

Обычно закон (16) комментируется как обеспечивающий заданный угол тангажа при демпфирующем свойстве обратной связи по ω_z .

Учитывая, что $\dot{\vartheta} = \omega_z$ и $\vartheta = \int_0^T \omega_z dt + C$, можно интерпретировать (16) как ПИ-регулятор (П — по ω_z , И — по ϑ), где $\vartheta_{\text{зад}}$ можно рассматривать, как постоянную интегрирования.

Важнейшим свойством тангажного автомата является стремление ω_z к нулю, а не отработка $\vartheta_{\text{зад}}$, как можно было бы предположить из названия этой структуры автомата. Таким образом, происходит автоматическая балансировка, так как при $\omega_z = 0$ сумма моментов вокруг оси Z, связанной с ЛА равна нулю.

Теперь можно констатировать, что отработка при помощи ПИ-регуляторов кусочно-постоянных значений угловых скоростей $\omega_x, \omega_y, \omega_z$ для движений (вращений) вокруг центра масс обеспечивает сбалансированность по моментам, а отработка также при помощи ПИ-регуляторов кусочно-постоянных значений V_ϵ и Ω приводит к силовой балансировке движений в вертикальной и горизонтальной плоскостях.

Вернёмся теперь к указанному выше преимуществу в отработке скоростей по сравнению с отработкой ускорений.

Дело в том, что кусочно-постоянные значения скоростей являются особыми точками, то есть положениями равновесия объекта управления ЛА, в то время как любые заданные по-

ложения $(H_{зад}, \Psi_{зад})$ положениями равновесия не являются, точно также, как и заданные углы $\Psi_{зад}, \vartheta_{зад}, \gamma_{зад}$.

Стабилизация заданных перегрузок была бы вообще безнадёжным делом, если бы не демпфирующие свойства собственно объекта управления – летательного аппарата.

Поэтому мощность (быстродействие) бортовых вычислителей, обеспечивающих стабилизацию перегрузки n_y , должны быть существенно выше, по сравнению с вычислителями, обеспечивающими стабилизацию скоростей.

1.1 Поддержание заданной высоты полёта

Цель. Ознакомиться с технологией математического моделирования динамических процессов при помощи имитационного моделирования.

Задачи. Создать математическую модель движения БЛА в соответствии с названием работы.

1.1.1 Теоретические вводные

$$H = y_2, \dot{H} = y_5 \quad (17)$$

Для горизонтального прямолинейного полёта с постоянной скоростью

$$\begin{aligned} n_{xT} &= 0; \\ \cos \theta &= \sqrt{1 - \frac{y_5^2}{y_4^2 + y_5^2 + y_6^2}}; \quad n_{yT} = \cos \theta + k_{ny}(y_{5z} - y_5); \\ n_{zT} &= 0; \end{aligned} \quad (18)$$

$$\begin{aligned} \dot{H}_3 &= k_h(H_3 - H) \\ y_{5_z} &= k_h(y_{2_z} - y_2) \end{aligned}$$

1.1.2 Порядок выполнения работы в среде SIMODO stars

Перед началом работы необходимо средствами операционной системы (ОС) создать рабочий каталог проекта, в котором будут располагаться файлы модели и сценарии моделирования.

Затем нужно запустить среду SIMODO stars и установить созданный каталог проекта в качестве рабочего каталога SIMODO stars. Краткое описание системы моделирования приведено в приложении А.

Далее в рабочем каталоге нужно создать файл модели и ввести в него формулы движения БЛА (12) и управляющего воздействия (17) и (18).

Примерный результат записи модели представлен на рисунке 4.

```
// Используем вспомогательный модуль для S-функции
import "support.scriptc0" type Support s;

// Движение центра масс
// Кинематика
y1' = y4;
y2' = y5;
y3' = y6;
// Динамика
y4' = g*y4 / sqrt(y4^2+y5^2+y6^2) * (nxt - y5 / sqrt(y4^2+y6^2) * nyt)
- g*y6 / sqrt(y4^2+y6^2) * nzt;
y5' = g*y5 / sqrt(y4^2+y5^2+y6^2) * nxt
+ g * sqrt(y4^2+y6^2) / sqrt(y4^2+y5^2+y6^2) * nyt
- g;
y6' = g*y6 / sqrt(y4^2+y5^2+y6^2) * (nxt - y5 / sqrt(y4^2+y6^2) * nyt)
+ g*y4 / sqrt(y4^2+y6^2) * nzt;

// Система управления
y5_z = kh * (y2_z - y2);
y5_z = s.limit(y5_z, y5_z_min, y5_z_max);
nxt = 0;
nyt = 1 + kny*(y5_z - y5);
nyt = s.limit(nyt, nyt_min, nyt_max);
nzt = 0;
```

Рисунок 4. Текст модели ЛР1 в среде SIMODO stars (файл «lab1.diff»)

На рисунке 5 приведён пример оформления сценария моделирования.

```
// Импорт модели БПЛА
import "lab1.diff" type LA;
// Создание объекта модели БПЛА и задание начальных условий
LA la = {
  g : 9.81,
  y1 : 0.0, y2 : 100.0, y3 : 0.0,
  y4 : 150.0, y5 : 0.0, y6 : 0.0,

  nyt : 1.0,

  y2_z : 250.0,
  kh : 0.2,
  kny : 1.0,

  y5_z_min : -70.0, y5_z_max : 15.0,
  nyt_min : -1.0, nyt_max : 5.0,
};
// Инициализация графиков
chart.init("ЛР1");
graph3d.init("ЛР1");
graph3d.addSeries("", graph3d.Mesh.Sphere);
// Определение функции обратного вызова (callback) для отображения
результатов моделирования
void onShow()
{
  chart.addPoint("y5", scene.t(), la.y5);
  chart.addPoint("y5_z", scene.t(), la.y5_z);
  chart.show();
  graph3d.addPoint("", la.y1, la.y2, la.y3);
}
// Инициализация сцены
scene.setIterationCallback(onShow, 100);
scene.setT(0.0);
scene.setDt(0.01);
scene.setTk(100.0);
// Добавление объектов на сцену
scene.add(la);
// Запуск моделирования
scene.start();
```

Рисунок 5. Запись сценария выполнения моделирования в среде SIMODO stars (файл «lab1.scriptc0»)

В результате выполнения моделирования должны получиться графики, изображённые на рисунках 6 и 7.

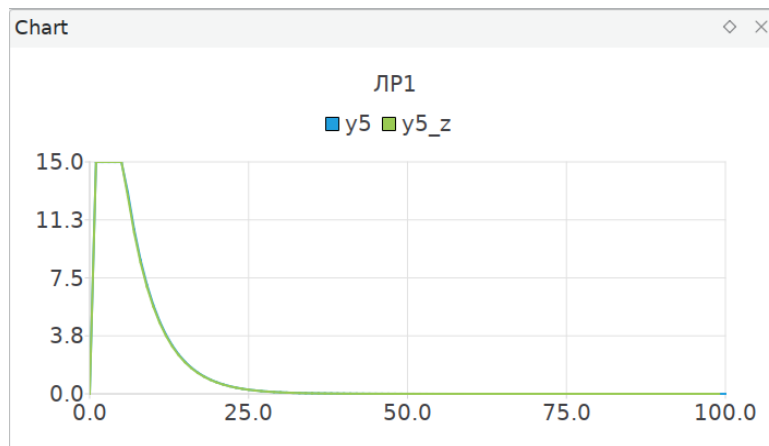


Рисунок 6. Вид двухмерного графика в среде SIMODO stars

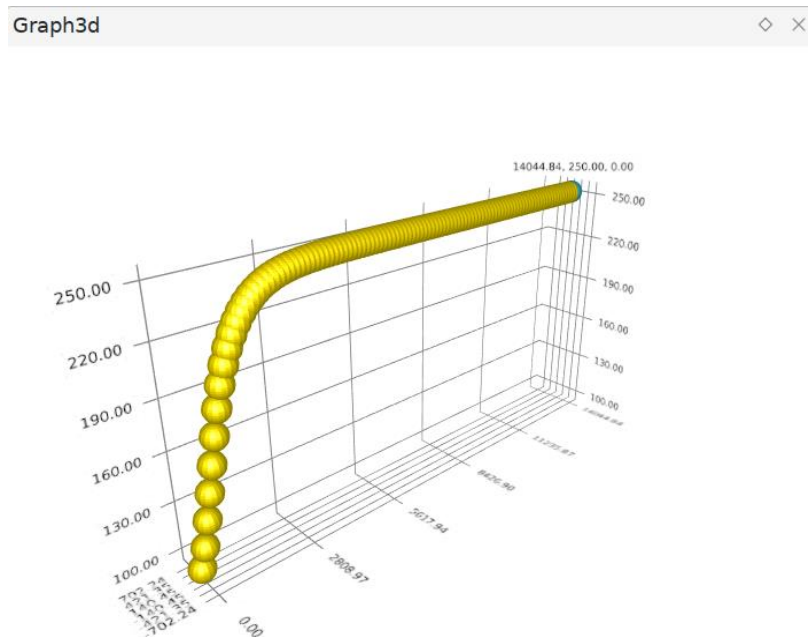


Рисунок 7. Траектория полёта БЛА на трёхмерном графике в среде SIMODO stars

1.1.3 Контрольные вопросы

- 1) Показать кинематические уравнения, входящие в математическую модель.
- 2) Показать динамические уравнения, входящие в математическую модель.
- 3) Показать управляющие воздействия.

1.2 Поддержание заданной скорости полёта

Цель. Понять отличия математических моделей.

Задачи. Создать математическую модель движения БЛА в соответствии с названием работы.

1.2.1 Теоретические вводные

$$\begin{aligned} n_{xT} &= k_V(V_z - V); \\ n_{yT} &= 1 + k_{ny}(y_{5z} - y_5); \\ n_{zT} &= 0; \end{aligned} \quad (19)$$

1.2.2 Порядок выполнения работы в среде SIMODO stars

Модель динамики движения БЛА не изменяется по сравнению с предыдущей ЛР, поэтому на рисунке 8 отображены только уравнения системы управления.

```
// Система управления
y5_z = kh * (y2_z - y2);
y5_z = s.limit(y5_z, y5_z_min, y5_z_max);
v = sqrt(y4^2+y5^2+y6^2);
nxt = k_v * (v_z - v);
nxt = s.limit(nxt, nxt_min, nxt_max);
nyt = 1 + kny*(y5_z - y5);
nyt = s.limit(nyt, nyt_min, nyt_max);
nzt = 0;
```

Рисунок 8. Запись уравнений системы управления в среде SIMODO stars

Перед выполнением моделирования необходимо установить необходимые начальные значения. На рисунке 9 показан примерный вид траектории движения БЛА, полученный в результате моделирования.

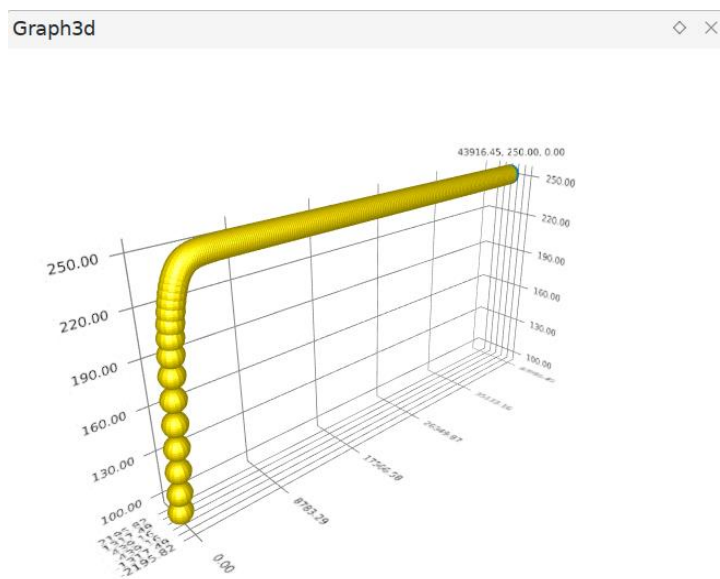


Рисунок 9. Вид траектории БЛА в среде SIMODO stars

1.2.3 Контрольные вопросы

- 1) Показать кинематические уравнения, входящие в математическую модель.
- 2) Показать динамические уравнения, входящие в математическую модель.
- 3) Показать управляющие воздействия.

1.3 Поддержание заданной боковой перегрузки

Цель. Понять отличия математических моделей.

Задачи. Создать математическую модель движения БЛА в соответствии с названием работы.

1.3.1 Теоретические вводные

$$\begin{aligned}
 n_{xT} &= k_V(V_z - V); \\
 n_{zT} &= \text{const}; \\
 n_{yT} &= 1 + k_{ny}(y_{5z} - y_5); \\
 y_{5z} &= k_h(y_{2z} - y_2)
 \end{aligned} \tag{20}$$

1.3.2 Порядок выполнения работы в среде SIMODO stars

Модель динамики движения БЛА не изменяется по сравнению с предыдущей ЛР, поэтому на рисунке 10 отображены только уравнения системы управления.

```
// Система управления
y5_z = kh * (y2_z - y2);
y5_z = s.limit(y5_z, y5_z_min, y5_z_max);
v = sqrt(y4^2+y5^2+y6^2);
nxt = k_v * (v_z - v);
nxt = s.limit(nxt, nxt_min, nxt_max);
nyt = 1 + kny*(y5_z - y5);
nyt = s.limit(nyt, nyt_min, nyt_max);
nzt = 0.5;
```

Рисунок 10. Запись уравнений системы управления в среде SIMODO stars

Перед выполнением моделирования необходимо установить необходимые начальные значения. На рисунке X показан примерный вид траектории движения БЛА, полученный в результате моделирования.

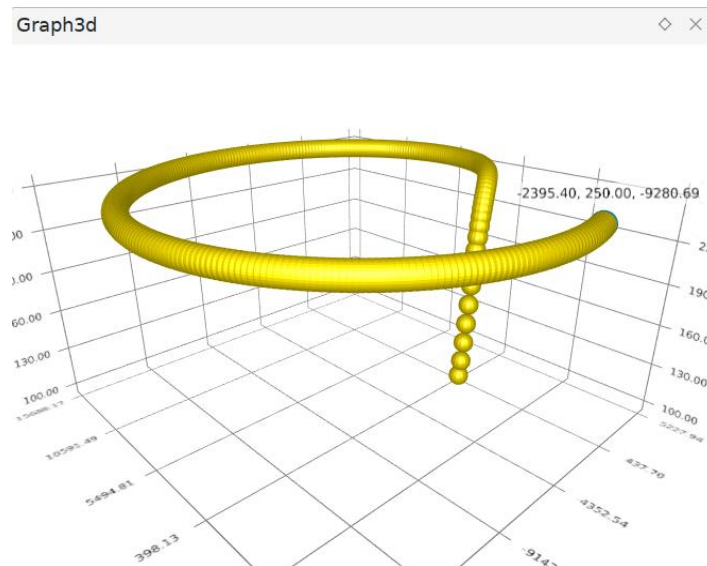


Рисунок 11. Вид траектории БЛА в SIMODO stars

1.3.3 Контрольные вопросы

- 1) Показать кинематические уравнения, входящие в математическую модель.
- 2) Показать динамические уравнения, входящие в математическую модель.
- 3) Показать управляющие воздействия.

1.4 Поддержание заданного курса полёта

Цель. Получить опыт создания алгоритмов и структуры систем стабилизации.

Задачи. Разработать алгоритм и структуру системы поддержания заданной скорости изменения высоты полета.

1.4.1 Теоретические вводимые

Для выхода на заданный курс полёта Ψ_z с некоторого исходного курса Ψ в соответствии со схемой на рисунке 1 формируется угловая скорость разворота Ω_z вектора скорости в горизонтальной плоскости V_z .

На рисунке 12 представлен фрагмент алгоритма управления БЛА в среде SIMODO stars.

```
SINPSI = y6 / Vg;
COSPSI = y4 / Vg;
ATAN2PSI = atan2(SINPSI, COSPSI);
PSIdir = (y6 >= 0. ? ATAN2PSI : (2*math.pi + ATAN2PSI));
OM = Kom*(psi_z - PSIdir);
nzt = Knzt*OM;
nzt = s.limit(nzt,nzt_min,nzt_max);
```

Рисунок 12. Фрагмент алгоритма расчёта моментов

Текущий угол курса Ψ обозначен переменной $\Psi_{id}r$. Угловая скорость разворота Ω_z – переменной ΩM .

1.4.2 Порядок выполнения работы в среде SIMODO stars

На рисунке 13 изображена траектория полёта БЛА.

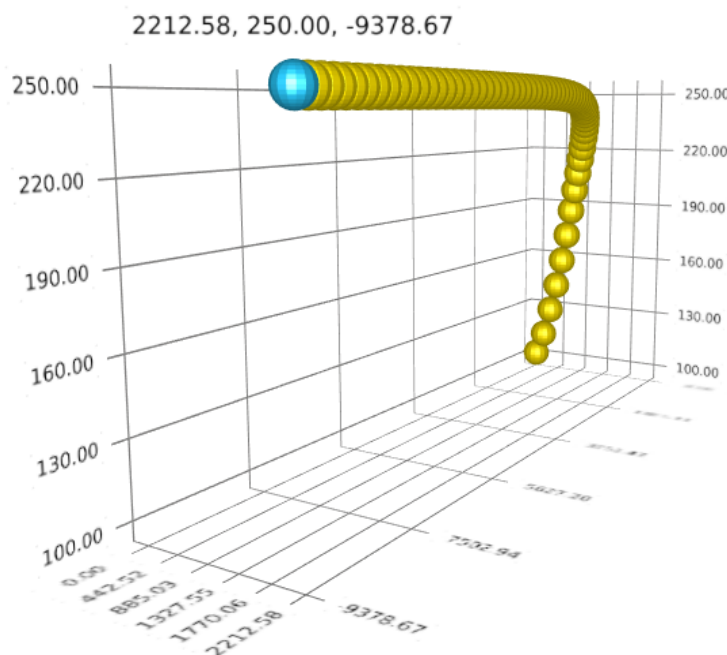


Рисунок 13. Вид траектории полёта БЛА в SIMODO stars

Файлы моделей и сценариев моделирования для SIMODO stars можно найти в вузовском репозитории исходных кодов по адресу <https://bmstu.codes/lxx/models/labs> [3].

1.4.3 Контрольные вопросы

- 1) Структура системы поддержания заданной скорости изменения высоты полета.
- 2) Что обеспечивает грубость автомата поддержания заданной вертикальной скорости?

1.5 Прохождение через окрестности заданной точки

Цель. Получение навыков в разработке алгоритмов и структуры системы управления.

Задачи. Разработать алгоритм и структуру системы поддержания заданной высоты полета.

1.5.1 Теоретические вводные

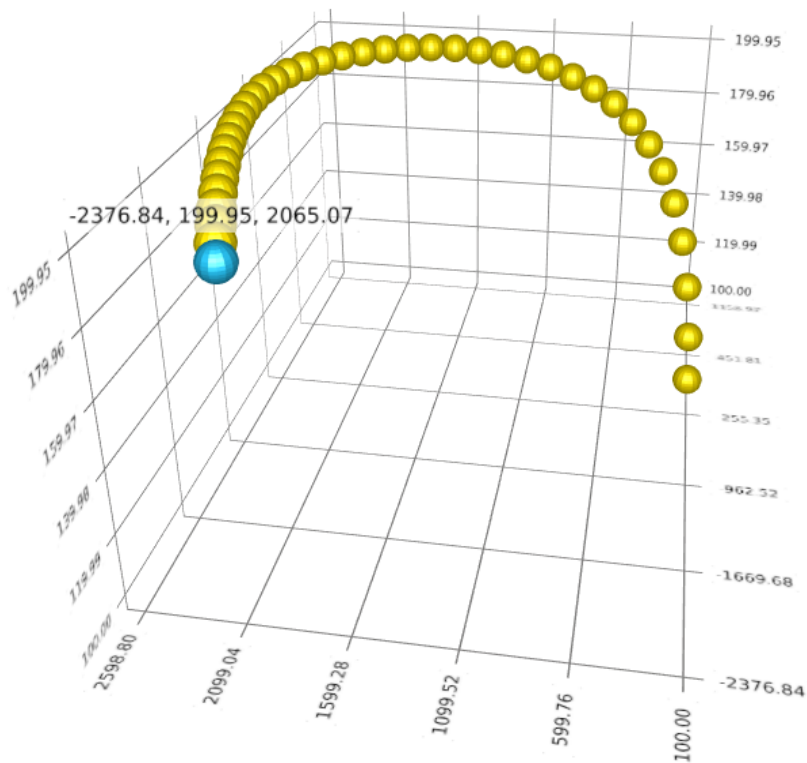
Для прохождения в окрестности заданной точки, которая будет рассматриваться как поворотный пункт маршрута (ППМ), выполняется поворот на некоторый заданный курс Ψ_z в

горизонтальной плоскости при помощи Ω_z (см. рисунок 1), а также достижение заданной высоты H_z при помощи отработки заданной скорости y_{5z} (см. рисунок 2).

Окрестность заданной точки можно задавать в виде цилиндра, чтобы облегчить расстановку ППМ с учётом реальных возможностей силовых установок БЛА.

1.5.2 Порядок выполнения работы в среде SIMODO stars

На рисунке 14 изображена траектория полёта БЛА.



1.6.1 Теоретические вводные

Маршрут представляет собой заранее расставленные поворотные пункты маршрута (ППМ); удалённые друг от друга в горизонте и по высоте с учётом ограничений на динамику БЛА.

Таким образом, прохождение по маршруту представляет собой многократное повторение решения предыдущей задачи: «Прохождение через окрестности заданной точки».

1.6.2 Порядок выполнения работы в среде SIMODO stars

На рисунке 15 изображен фрагмент алгоритма с определением ППМ в виде вектора параметров, а также вспомогательных переменных.

```
auto ppm = [
    // xc, y2_z, zc, V_z,
    [-2500.0, 200.0, 2000.0, 200.0, ],
    [-2500.0, 500.0, -2000.0, 200.0, ],
    [2500.0, 100.0, 2000.0, 200.0, ],
    [-5500.0, 200.0, 2000.0, 200.0, ],
    [-500.0, 500.0, -2000.0, 200.0, ],
    [2500.0, 100.0, 2000.0, 200.0, ],
    [-4500.0, 200.0, 2000.0, 200.0, ],
    [-2500.0, 500.0, -2000.0, 200.0, ],
    [2500.0, 100.0, 2000.0, 200.0, ],
];
int ppm_current_index = 0;
int ppm_max_index = sys.dimension_size(ppm,0)-1;
float Rtarget = 50.0;
```

Рисунок 15. Задание ППМ в виде вектора параметров

На рисунке 16 представлен вариант определения функции для вычисления прохождения БЛА по маршруту.

```
void onIteration()
{
    if (math.sqrt((la.xc-la.y1)^2 + (la.zc-la.y3)^2) < Rtarget)
    {
        if (ppm_current_index == ppm_max_index)
        {
            sys.print("Цель достигнута, время: "+scene.t()+" c" );
            scene.setTk(scene.t());
        }
        else
        {
            sys.print("Точка "+ppm_current_index+" достигнута, время: "+scene.t()+" c" );
            ppm_current_index ++;
            la.xc = ppm[ppm_current_index,0];
            la.y2_z = ppm[ppm_current_index,1];
            la.zc = ppm[ppm_current_index,2];
            la.V_z = ppm[ppm_current_index,3];
        }
    }
}
// Инициализация сцены
scene.setEachIterationCallback(onIteration);
```

Рисунок 16. Алгоритм вычисления ППМ на основе матрицы

На рисунке изображена панель сообщений времени проведения моделирования, куда выводятся строки, переданные в метод `sys.print`.

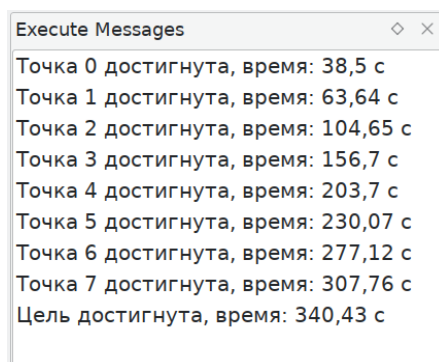


Рисунок 17. Панель сообщений времени выполнения моделирования (результат вызова метода `sys.print`)

На рисунке 18 изображена траектория полёта БЛА.

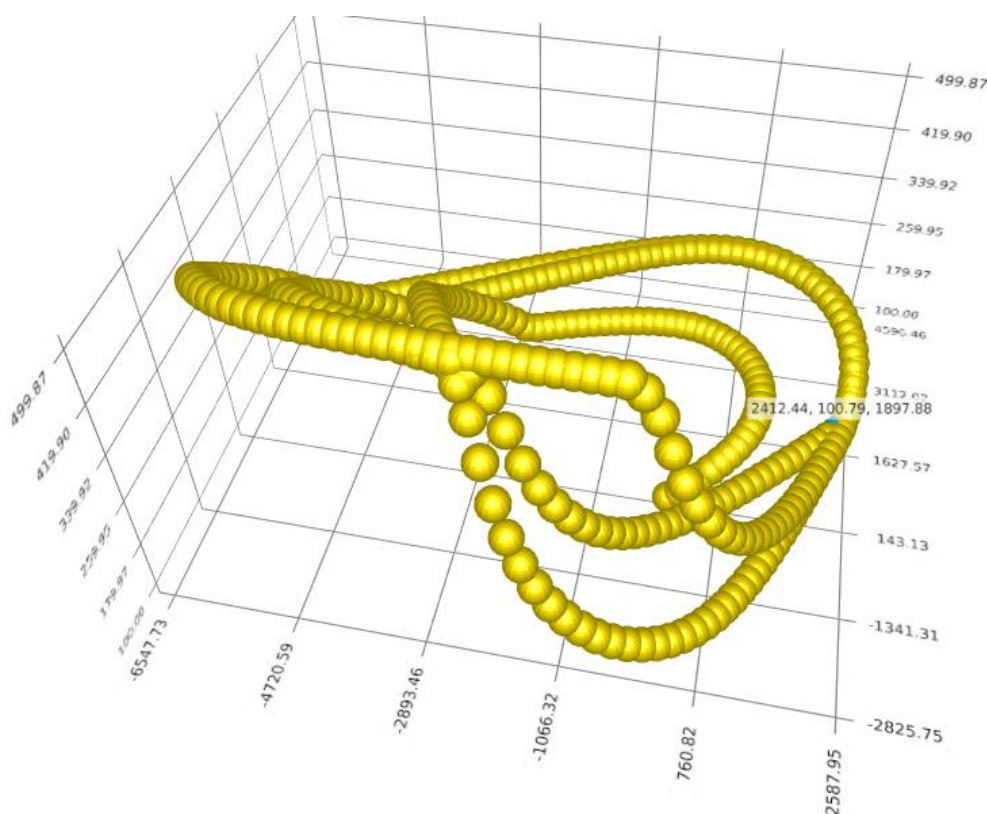


Рисунок 18. Вид траектории полёта БЛА в SIMODO stars

Файлы моделей и сценариев моделирования для SIMODO stars можно найти в вузовском репозитории исходных кодов по адресу <https://bmstu.codes/lxx/models/labs> [3].

1.6.3 Контрольные вопросы

- 1) Структура системы поддержания заданного курса полета.
- 2) Что обеспечивает грубость автомата поддержания заданного курса полета.

2 Алгоритмы и структуры системы управления полетом БЛА

2.1 Автоматический взлёт

Цель. Получение навыков в разработке алгоритмов и структуры системы управления.

Задачи. Разработать алгоритм и структуру фрагмента системы управления полетом.

2.1.1 Теоретические вводные

Моделирование автоматического взлёта выполняется следующим образом.

Самолёт разгоняется по взлётно-посадочной полосе (ВВП) на высоте $H = 0$ до скорости отрыва от ВВП (например, до $V_{отр} = 80 \text{ м/с}$). Это было разобрано в задаче «Поддержание заданной скорости полёта».

Далее устанавливается заданная высота, после достижения которой выполняется полёт по маршруту.

2.1.2 Порядок выполнения работы в среде SIMODO stars

На рисунке 19 изображена траектория полёта БЛА.

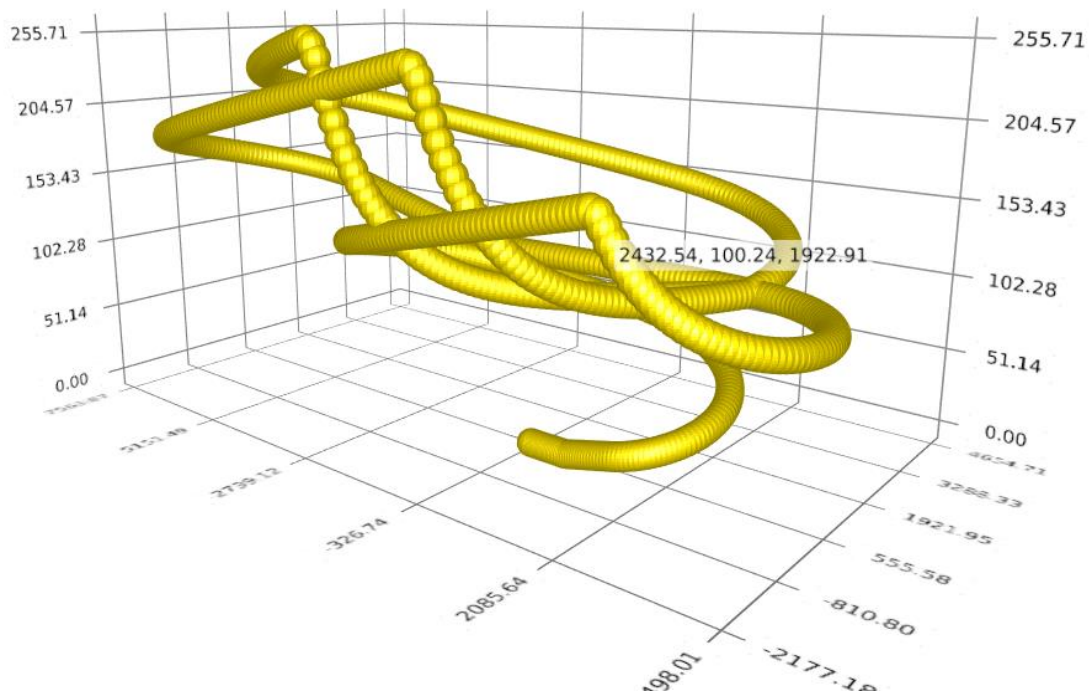


Рисунок 19. Вид траектории полёта БЛА

Файлы моделей и сценариев моделирования для SIMODO stars можно найти в вузовском репозитории исходных кодов по адресу <https://bmstu.codes/lxx/models/labs> [3].

2.1.3 Контрольные вопросы

- 1) Структура системы, обеспечивающей попадание в окрестность заданной точки.
- 2) Что обеспечивает грубость фрагмента системы управления полетом.

2.2 Автоматическая посадка с выравниванием

Цель. Получение навыков в разработке алгоритмов и структуры системы управления.

Задачи. Разработать алгоритм и структуру фрагмента системы управления полетом.

2.2.1 Теоретические вводные

Моделирование посадки с выравниванием выполняется следующей последовательностью ранее решаемых задач.

Достижение точки начала снижения по глиссаде как точки маршрута с заданным курсом, на заданной высоте и с заданной скоростью. Далее смена эшелона до высоты выравнивания с ограничением вертикальной скорости снижения. После достижения высоты выравнивания сбрасывается скорость полёта до посадочной скорости. Посадочная скорость меньше эволютивной скорости (минимальной скорости, на которой возможен горизонтальный полёт), поэтому при достижении посадочной скорости подъёмная сила уже меньше веса БЛА, что моделируется при помощи (21).

$$n_{yT} = 0,8 + k_{y5}(y_{5\text{зад}} - y_5), \quad (21)$$

где $y_{5\text{зад}} = 0$.

2.2.2 Порядок выполнения работы в среде SIMODO stars

Файлы моделей и сценариев моделирования для SIMODO stars можно найти в вузовском репозитории исходных кодов по адресу <https://bmstu.codes/lxx/models/labs> [3].

2.2.3 Контрольные вопросы

- 1) Структура системы, обеспечивающей полет по маршруту.
- 2) Что обеспечивает грубость фрагмента системы управления полетом.

2.3 Посадка в заданную точку

Цель. Получение навыков в разработке алгоритмов и структуры системы управления.

Задачи. Разработать алгоритм и структуру фрагмента системы управления полетом.

2.3.1 Теоретические вводные

Для реализации моделирования посадки в заданную точку решается последовательность задач.

Сначала выполняется заход на посадку, т.е. попадание в точку начала снижения. Затем выполняется снижение со стабилизацией посадочной скорости V_{noc} . Управление по вертикали для попадания в заданную точку на поверхности земли выполняется при помощи стабилизации вертикальной скорости $V_y = y_{53}$, которая и обеспечивает отработку заданного угла наклона траектории, см. рисунок 20 и (22).

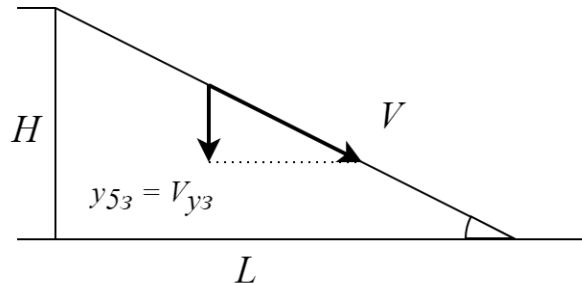


Рисунок 20. Схема получения заданного угла наклона траектории по заданной вертикальной скорости

$$\begin{aligned} TET_{\text{зад}} &= \arctg \frac{H}{L}; \\ TET &= \arcsin \frac{V_y}{V}; \end{aligned} \quad (22)$$

2.3.2 Порядок выполнения работы в среде SIMODO stars

Файлы моделей и сценариев моделирования для SIMODO stars можно найти в вузовском репозитории исходных кодов по адресу <https://bmstu.codes/lxx/models/labs> [3].

2.3.3 Контрольные вопросы

- 1) Структура системы, обеспечивающей автоматическую посадку с «выравниванием».
- 2) Что обеспечивает грубость фрагмента системы управления полетом.

2.4 Взлёт, прохождение ППМ и посадка

Цель. Получение навыков в разработке алгоритмов и структуры системы управления.

Задачи. Разработать алгоритм и структуру фрагмента системы управления полетом.

2.4.1 Теоретические вводные

Последовательность ранее решённых задач.

2.4.2 Порядок выполнения работы в среде SIMODO stars

Файлы моделей и сценариев моделирования для SIMODO stars можно найти в вузовском репозитории исходных кодов по адресу <https://bmstu.codes/lxx/models/labs> [3].

2.4.3 Контрольные вопросы

- 1) Структура системы, обеспечивающей автоматическую посадку в окрестность заданной точки на ВПП.
- 2) Что обеспечивает грубость фрагмента системы управления полетом.

2.5 Автоматическое уклонение от потенциальной угрозы

Цель. Получение навыков в разработке алгоритмов и структуры системы управления.

Задачи. Разработать алгоритм и структуру фрагмента системы управления полетом.

2.5.1 Теоретические вводные

Выполняется маневрирование двух БЛА, атакующих друг друга при помощи ракет воздух-воздух. Оба БЛА пытаются уклониться от атакующих их ракет. Манёвр уклонения выполняется по следующему алгоритму.

Предполагается, что на борту БЛА есть информация о взаимном расположении объектов (БЛА и ракета).

В момент пуска ракеты БЛА наводится на ракету, как на подвижную точку цели. После сближения с целью на заданное расстояние происходит переключение на уклонение от движущейся цели, т.е. меняется знак Ω_z . Заданное расстояние, на котором сближение заменяется на уклонение, подбирается в процессе моделирования таким образом, чтобы максимизировать угловую скорость линии визирования от ракеты на БЛА.

2.5.2 Порядок выполнения работы в среде SIMODO stars

На рисунке 21 изображена траектория полёта БЛА.

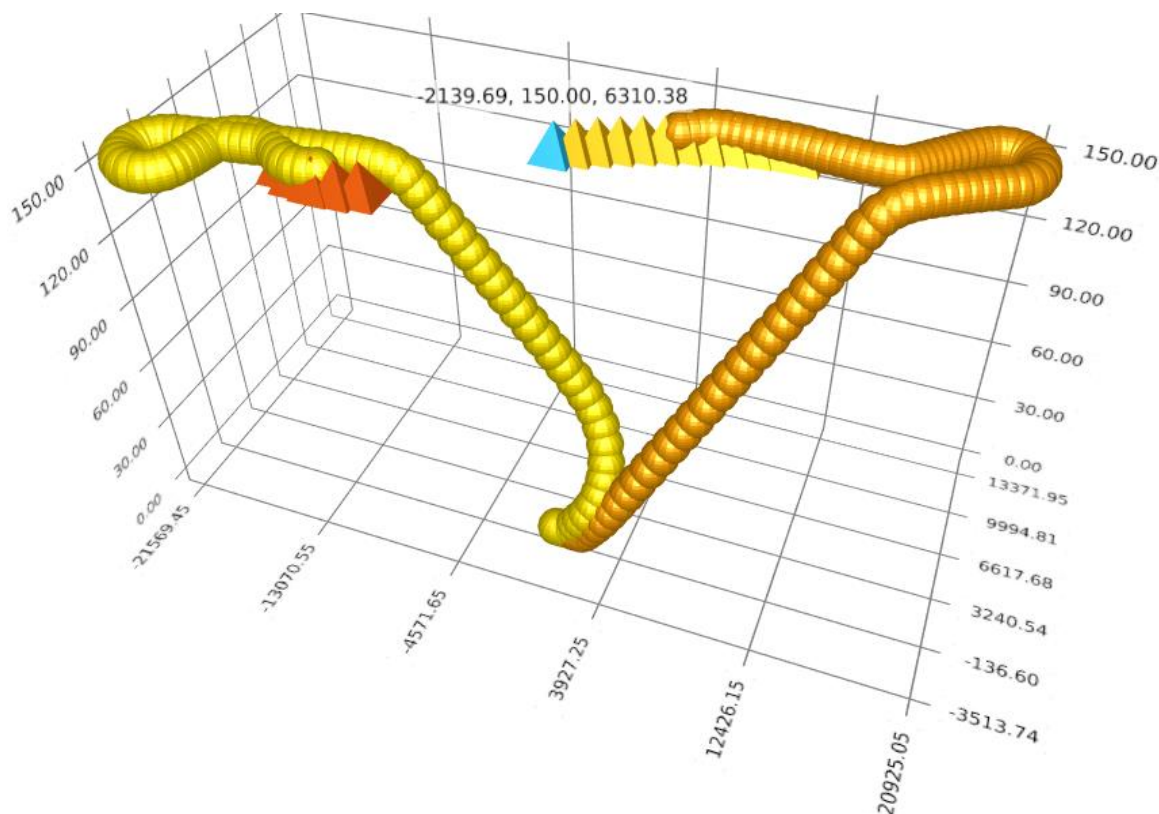


Рисунок 21. Вид траектории полёта БЛА в SIMODO stars

Файлы моделей и сценариев моделирования для SIMODO stars можно найти в вузовском репозитории исходных кодов по адресу <https://bmstu.codes/lxx/models/labs> [3].

2.5.3 Контрольные вопросы

2.6 Сброс объекта на заданный участок

Цель. Получение навыков в разработке алгоритмов и структуры системы управления.

Задачи. Разработать алгоритм и структуру фрагмента системы управления полетом.

2.6.1 Теоретические вводные

Выполняется горизонтальный полёт в заданную точку, затем на сцену вызывается отделившийся от БЛА объект, свободно падающий на землю. Начальное значение скорости отделившегося от БЛА объекта совпадает со скоростью БЛА. Далее падение объекта моделируется при помощи интегрирования аналогичной системы дифференциальных уравнений при

$$\begin{aligned} n_{xT} &= k_{xв} \frac{v^2}{mg}; \\ n_{yT} &= 0; \\ n_{zT} &= 0; \end{aligned}$$

$$n_{xT} = -1 \div 0, k_{xв} < 0;$$

В результате подбора $k_{хв} = -0,3$.

После достижения объектом высоты $y_2 = 0$ моделирование его движения прекращается. Моделирование движения носителя продолжается.

2.6.2 Порядок выполнения работы в среде SIMODO stars

На рисунке 22 изображена цель и траектории полёта БЛА и сбрасываемого объекта.

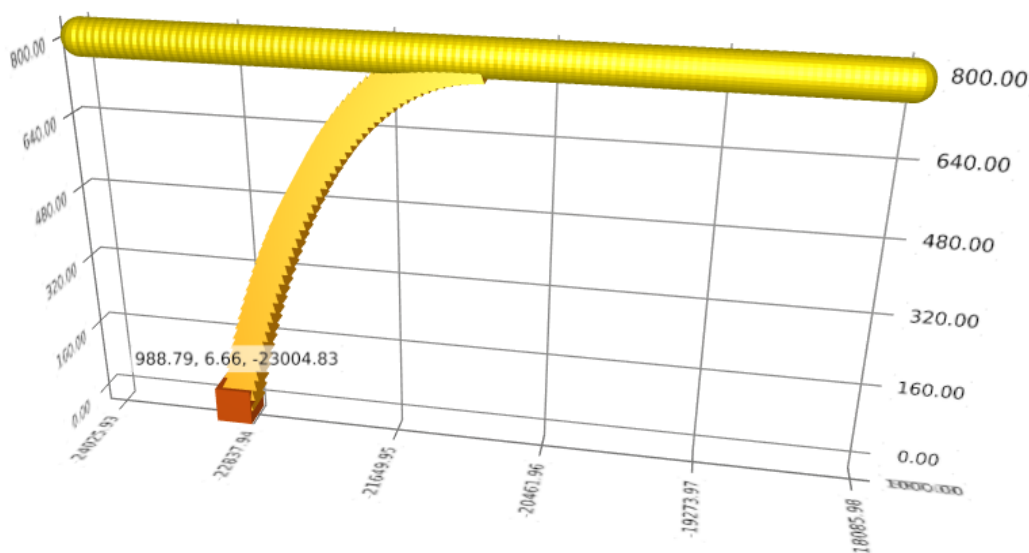


Рисунок 22. Вид цели и траектории полёта БЛА и сбрасываемого объекта в SIMODO stars

Файлы моделей и сценариев моделирования для SIMODO stars можно найти в вузовском репозитории исходных кодов по адресу <https://bmstu.codes/lxx/models/labs> [3].

2.6.3 Контрольные вопросы

2.7 Моделирование группового взаимодействия

Цель. Получение навыков в разработке алгоритмов и структуры системы управления.

Задачи. Разработать алгоритм и структуру фрагмента системы управления полетом.

2.7.1 Теоретические вводные

Моделируется следующая сцена. Ведущий пилотируемый самолёт сопровождает четырёх ведомых БЛА, которые расположены в ряд на расстоянии 3 км друг от друга на одной линии, опережая ведущего на расстояние 3 км (см. рисунок 23).

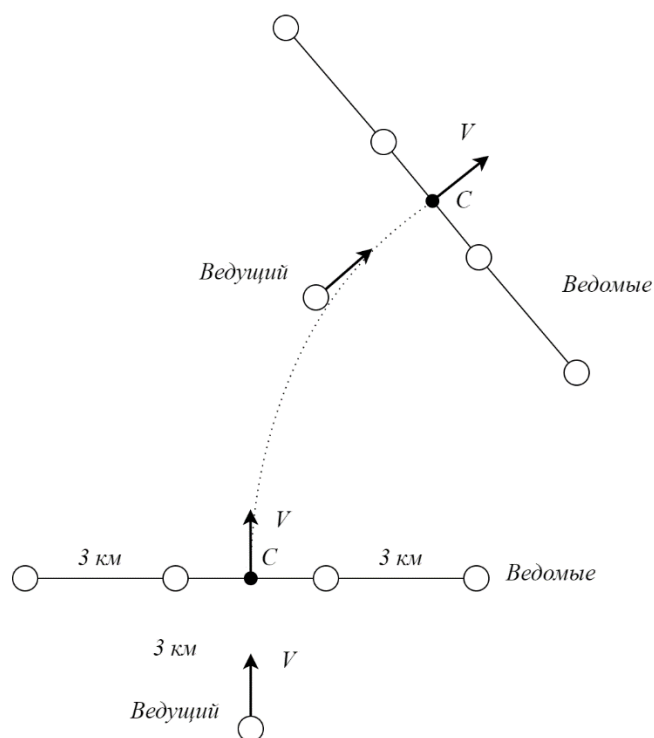


Рисунок 23. Схема полёта пилотируемого ЛА с группой ведомых БЛА

Центр группы ведомых (точка С) повторяет манёвры ведущего и дистанцию 3 км. Ведомые занимают свои места в строю относительно точки центра группы ведомых (точка С).

В заданный момент времени появляются четыре цели: две крылатые ракеты и два корабля. Два ведомых БЛА атакуют при помощи ракет воздух-воздух крылатые ракеты. Два других БЛА атакуют при помощи бомб корабли.

Всего в сцене участвуют 13 объектов.

2.7.2 Порядок выполнения работы в среде SIMODO stars

На рисунке 24 изображена траектория полёта БЛА.

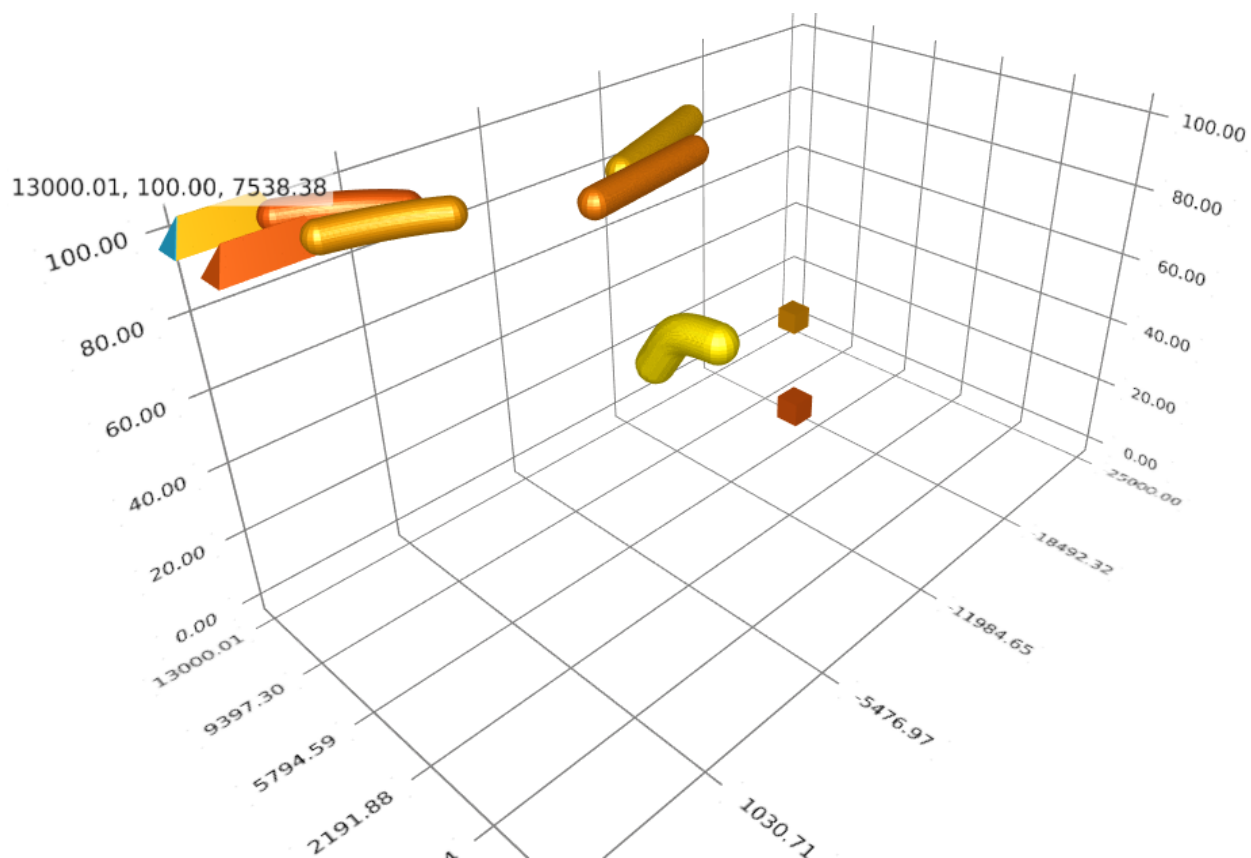


Рисунок 24. Вид траектории полёта БЛА в SIMODO stars

Файлы моделей и сценариев моделирования для SIMODO stars можно найти в вузовском репозитории исходных кодов по адресу <https://bmstu.codes/lxx/models/labs> [3].

2.7.3 Контрольные вопросы

Приложение А. Краткое описание SIMODO stars

А.1 Общее описание

Адаптивная система моделирования (АСМ) – совокупность программных компонентов, обеспечивающих автоматизацию разработки предметно-ориентированных языков (ПОЯ) для различных предметных областей, а также анализ и выполнение моделей, описанных на этих ПОЯ, в том числе в распределённой среде.

SIMODO stars – это прототип адаптивной системы моделирования, разрабатываемой в МГТУ им. Н.Э. Баумана. SIMODO stars создана для отработки основных принципов построения универсальных интерпретаторов с ПОЯ на примере решения обыкновенных дифференциальных уравнений, а также интегрированной среды разработки, поддерживающей автоматизацию создания ПОЯ.

В SIMODO stars реализован императивный язык для описания алгоритмов «scriptc0» и декларативный язык для описания системы обыкновенных дифференциальных уравнений «diff», а также одноимённая интегрированная среда разработки.

Описание процесса моделирования в SIMODO stars предусматривает как минимум два этапа:

- 1) формирование моделей;
- 2) создание сценария выполнения моделирования, в котором определяются: состав объектов сцены, их начальные условия, настройка отображения информации и запуск на моделирование.

Такое разделение обусловлено требованием к АСМ по изоляции описания модели, которое выполнено на языке, близком к записи системы дифференциальных уравнений, от собственно моделирования, в котором могут принимать участие объекты, как одной, так и нескольких моделей. Такой подход упрощает формирование сложных сцен, которые могут состоять из большого количества объектов, как одной, так и разных моделей.

А.2 Интегрированная среда разработки SIMODO stars

На рисунке 12 представлен внешний вид интегрированной среды разработки SIMODO stars с обозначением основных элементов интерфейса (детали интерфейса в конкретной установке SIMODO stars могут отличаться).

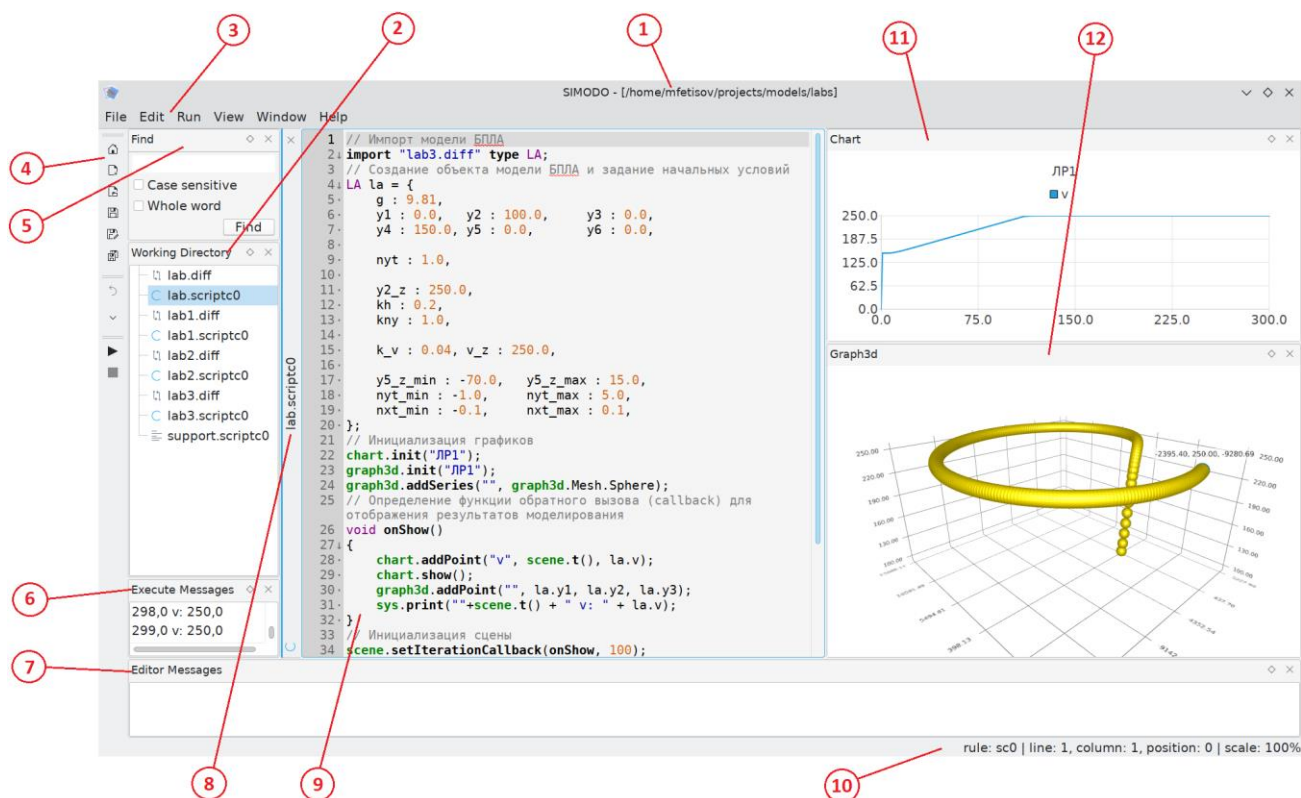











Рисунок 25. Окно интегрированной среды разработки SIMODO stars

В таблице 1 дано краткое описание основных элементов пользовательского интерфейса интегрированной среды разработки SIMODO stars.

Таблица 1. Основные элементы главного окна интегрированной среды разработки SIMODO stars

№№	Название элемента	Краткое описание элемента
1.	Заголовок окна	В заголовке окна отображается путь к каталогу проекта (рабочему каталогу).
2.	Панель файлов рабочего каталога	Панель файлов рабочего каталога отражает структуру каталогов, начиная с рабочего. В панели можно выполнять некоторые действия над файлами и каталогами. Доступно локальное меню при нажатии на правую кнопку мышки. Активация панели производится в меню View / Docked windows / Working directory.
3.	Главное меню	Главное меню является основным местом вызова команд в интегрированной среде разработки SIMODO stars. Для многих команд доступна комбинация клавиш для быстрой активации. Эти комбинации отображаются справа от названия команды в меню.

№№	Название элемента	Краткое описание элемента
4.	Панель команд	<p>На панели команд располагаются часто используемые команды. Вот некоторые из них (изображения значков в конкретной установке SIMODO stars могут отличаться):</p> <p> - сменить каталог проекта (рабочий каталог), повторяет команду меню File / Set working directory;</p> <p> - сохранить редактируемый файл (File / Save);</p> <p> - сохранить все несохранённые файлы (File / Save All);</p> <p> - отменить последнее изменение в редактируемом файле (Edit / Undo);</p> <p> - вернуть последнее отменённое изменение в редактируемом файле (Edit / Redo);</p> <p> - открыть / скрыть панель поиска в редактируемом файле (Edit / Find);</p> <p> - открыть / скрыть панель поиска и замены в редактируемом файле (Edit / Replace);</p> <p> - запустить редактируемый файл на выполнение (Run / Run);</p> <p> - остановить выполнение (Run / Stop).</p> <p>Активация элементов этой панели производится в меню View / Toolbars.</p>
5.	Панель поиска	Команда на активацию / деактивацию панели поиска находится в меню Edit / Find. Помимо панели поиска есть панель поиска и замены, которая активируется в меню Edit / Replace.
6.	Панель сообщений выполнения	<p>На эту панель выводятся сообщения, формируемые методом sys.print (см. А.5.3).</p> <p>Активация панели производится в меню View / Docked windows / Execute messages.</p>
7.	Панель сообщений редактора	<p>На эту панель выводятся замечания, генерируемые при фоновом анализе редактируемого файла, а также при исполнении сценария. При активации строки с сообщением (двойным нажатии левой кнопки мыши) в редакторе откроется файл, в котором было обнаружено замечание, а курсор установится на указанной в замечании позиции.</p> <p>Активация панели производится в меню View / Docked windows / Editor messages.</p>
8.	Имя редактируемого файла	На этой панели отображаются файлы, открытые в редакторе. Редактируемый файл выделяется.
9.	Окно с редактируемым файлом	Некоторые команды редактора представлены в меню Edit. Также доступны наиболее распространённые команды редактирования текста.

№№	Название элемента	Краткое описание элемента
10.	Строка статуса	В строке статуса отображается строка и позиция курсора в редактируемом файле. А также другая информация.
11.	Панель двухмерного графика	Панель отображает двухмерный график, данные для которого передаются с помощью встроенного модуля chart (см. А.5.4). Для этой панели доступно локальное меню, которое открывается при нажатии на правую кнопку мыши.
12.	Панель трёхмерного графика	Панель отображает двухмерный график, данные для которого передаются с помощью встроенного модуля graph3d (см. А.5.5). Трёхмерный график можно крутить, двигая мышку и зажав правую кнопку. Увеличивать / уменьшать масштаб можно, покрутив колёсико мышки. Левой кнопкой мышки можно указать точку и рядом с ней отобразятся значения, по которым она была сформирована. Для выбора цветовой схемы нужно зайти в меню View / Graph3d theme.

А.3 Язык описания алгоритмов scriptc0

Язык scriptc0 является интерпретируемым языком со строгой типизацией, предназначен для описания сценариев и алгоритмов. Он повторяет многие элементы языка С. Реализованы следующие элементы:

- 1) Комментарии в стиле языка С («//...» и «/*...*/»).
- 2) Строковые, числовые (целые и вещественные) и бинарные («true» и «false») константы.
- 3) Выражения с операциями (в порядке возрастания приоритета):
 - а) тернарная операция вида <логическое выражение> «?» <выражение1> «:» <выражение2>, где <выражение1> будет выполнено в случае истинности значения <логическое выражение>, а выражение <выражение2> - ложности;
 - б) логическое «или»: «||»;
 - в) логическое «и»: «&&»;
 - г) сравнение: «==», «!=», «>», «>=», «<», «<=»;
 - д) арифметические операции «+» и «-»;
 - е) арифметические операции «*», «/» и «%» (остаток от целочисленного деления);
 - ж) возведение в степень «^»;
 - з) унарные операции «+», «-» и «!» (логическое «нет»);
 - и) целочисленные префиксные и постфиксные итерации «++» и «--»;

- к) указание элемента типа «.» и значение массива по индексу «[<перечень индексов>]»;
 - л) вызов функции;
 - м) скобки «(<выражение>)».
- 4) Объявление переменной, при котором необходимо указать один из следующих типов:
 - а) целый: «int»;
 - б) вещественный: «float»;
 - в) строковый: «string»;
 - г) логический: «bool»;
 - д) тип, определённый в операторе «import»;
 - е) автоматическое определение типа по типу значения выражения инициализации: «auto».
 - 5) При объявлении переменной можно сразу её инициализировать.
 - 6) При объявлении переменной можно использовать массивы (кроме типа, определённого в операторе «import»).
 - 7) Присваивание;
 - 8) Оператор импорта;
 - 9) Условный оператор;
 - 10) Операторы циклов while, do-while и for;
 - 11) И другие.

В описании порядка выполнения лабораторных работ приводятся примеры формирования сценариев на языке scriptc0.

А.4 Язык описания дифференциальных уравнений diff

Язык diff является описательным (декларативным) интерпретируемым языком, предназначен для описания моделей динамических систем. Он состоит из следующих операторов:

- 1) импорта, аналогичного scriptc0;
- 2) присваивания, который имитирует простое уравнение;
- 3) производной, который имеет запись, похожую на присваивание, но перед знаком «=» нужно ставить знак одинарного апострофа «'».

В описании порядка выполнения лабораторных работ приводятся примеры описания моделей на языке diff.

A.5 Встроенные модули

A.5.1 math

Встроенный модуль `math` предназначен для инкапсуляции математических функций. В таблице 2 приводится перечень методов встроенного модуля `math`.

Таблица 2 – Перечень методов встроенного модуля `math`

Метод	Описание
<code>sin(float x)</code>	Метод возвращает значение синуса для заданного аргумента x
<code>cos(float x)</code>	Метод возвращает значение косинуса для заданного аргумента x
<code>tan(float x)</code>	Метод возвращает значение тангенса от заданного аргумента x
<code>asin(float x)</code>	Метод возвращает значение арксинуса от заданного аргумента x
<code>acos(float x)</code>	Метод возвращает значение арккосинуса от заданного аргумента x
<code>atan(float x)</code>	Метод возвращает значение арктангенса от заданного аргумента x
<code>sqrt(float x)</code>	Метод возвращает значение квадратного корня от заданного аргумента x
<code>exp(float x)</code>	Метод возвращает значение экспоненциальной функции от заданного аргумента x
<code>ln(float x)</code>	Метод возвращает значение натурального логарифма от заданного аргумента x
<code>abs(float x)</code>	Метод возвращает значение модуля заданного аргумента x
<code>sign(float x)</code>	Метод возвращает -1.0 при отрицательном заданном аргументе x и 1.0 во всех остальных случаях
<code>atan2(float y, float x)</code>	Метод возвращает значение арктангенса от заданного аргумента y/x
<code>srand(float seed)</code>	Метод устанавливающий зерно генератора псевдослучайных чисел
<code>rand()</code>	Метод возвращающий псевдослучайное целочисленное значение в диапазоне от 0 до как минимум 32767
<code>pi</code>	Константа со значением числа Пи

A.5.2 scene

Встроенный модуль scene предназначен для реализации функций численного решения дифференциальных уравнений. В таблице 3 приводится перечень методов встроенного модуля scene.

Таблица 3 – Перечень методов встроенного модуля scene

Метод	Описание
<code>setT(float x)</code>	Метод устанавливает текущее время моделирования
<code>t()</code>	Метод возвращает текущее время моделирования
<code>setTk(float x)</code>	Метод устанавливает время окончания моделирования
<code>tk()</code>	Метод возвращает время окончания моделирования
<code>setDt(float x)</code>	Метод устанавливает шаг моделирования
<code>dt()</code>	Метод возвращает шаг моделирования
<code>setIterationCallback(function f, int n)</code>	Метод устанавливает функцию, которая будет вызываться каждые n итераций моделирования
<code>setEachIterationCallback(function f)</code>	Метод устанавливает функцию, которая будет вызываться каждую итерацию моделирования
<code>setRealtimeModeEnabled(bool x)</code>	Метод включает или выключает режим реального времени, в котором время между итерациями моделирования приблизительно равно шагу моделирования
<code>realtimeModeEnabled()</code>	Метод возвращает статус режима реального времени
<code>add(object x)</code>	Метод добавляет объект x на сцену
<code>remove(object x)</code>	Метод удаляет объект x из сцены
<code>start()</code>	Метод начинает моделирование

A.5.3 sys

Встроенный модуль sys предназначен для инкапсуляции системных функций. В таблице 4 приводится перечень методов встроенного модуля sys.

Таблица 4 – Перечень методов встроенного модуля sys

Метод	Описание
<code>print(any x)</code>	Метод выводит на устройство отображения значение аргумента.
<code>typeof(any x)</code>	Возвращает строку с наименованием типа переменной <i>x</i> .
<code>sizeof(any x)</code>	Для скалярной переменной возвращает количество занимаемых байт. Для массива – количество всех элементов. Для структуры – количество элементов первого уровня.
<code>dimensions(any x)</code>	Возвращает размерность массива.
<code>dimension_size(any x, int index)</code>	Возвращает количество элементов для заданной размерности массива.
<code>version</code>	Определяет структуру текущей версии SIMODO stars. Содержит следующие элементы: <i>major</i> , <i>middle</i> , <i>minor</i> .
<code>time()</code>	Возвращает целое число внутреннего времени (не связано со временем сцены).

A.5.4 chart

Встроенный модуль `chart` предназначен для работы с панелью двумерных графиков интегрированной среды разработки SIMODO stars. В таблице 5 приводится перечень методов встроенного модуля `chart`.

Таблица 5 – Перечень методов встроенного модуля chart

Метод	Описание
<code>init(string title)</code>	Инициализация графика с заданием его наименования
<code>addSeries(string sername, int style)</code>	Определяет стиль серии точек для графика
<code>addPoint(string sername, float x, float y)</code>	Добавляет точку для заданной серии графика.
<code>show()</code>	Отображает изменения в графике

Метод	Описание
Style	Константный объект, содержащий следующие элементы: Line – линейная аппроксимация Spline – аппроксимация сплайнами Scatter – отображение заданных точек

При запуске сценария, в котором используется встроенный модуль `chart`, в утилите `simfuze`, выполнение завершится с ошибкой.

A.5.5 graph3d

Встроенный модуль `graph3d` предназначен для работы с панелью трёхмерных графиков интегрированной среды разработки `SIMODO stars`. В таблице 6 приводится перечень методов встроенного модуля `graph3d`.

Таблица 6 – Перечень методов встроенного модуля `graph3d`

Метод	Описание
<code>init(string title)</code>	Инициализация графика с заданием его наименования
<code>addSeries(string sername, int mesh)</code>	Определяет сетку серии точек для графика
<code>addPoint(string sername, float x, float y, float z)</code>	Добавляет точку для заданной серии графика.
<code>setPointsCount(string sername, int points)</code>	Определяет количество последних точек, отображаемых на графике. При передаче значения 0 прекращает контролировать количество точек в серии.
Mesh	Константный объект, содержащий следующие элементы: <code>Bar</code> , <code>Cube</code> , <code>Pyramid</code> , <code>Cone</code> , <code>Cylinder</code> , <code>BevelBar</code> , <code>BevelCube</code> , <code>Sphere</code> , <code>Minimal</code> , <code>Arrow</code> , <code>Point</code> .

При запуске сценария, в котором используется встроенный модуль `graph3d`, в утилите `simfuze`, выполнение завершится с ошибкой.

Литература

- 1 Самостоятельно устанавливаемый образовательный стандарт высшего образования МГТУ им. Н.Э. Баумана (на основе ФГОС 3++) / уровень высшего образования бакалавриат / направление подготовки 09.03.01 «Информатика и вычислительная техника». – Москва, 2019.
- 2 Самостоятельно устанавливаемый образовательный стандарт высшего образования МГТУ им. Н.Э. Баумана (на основе ФГОС 3++) / уровень высшего образования бакалавриат / направление подготовки 09.03.03 «Прикладная информатика». – Москва, 2019.
- 3 Примеры выполнения ЛР по дисциплине «Управляющие комплексы беспилотных летательных аппаратов». [Электронный ресурс]. URL: <https://bmstu.codes/lxx/models/labs> (дата обращения: 07.08.2022).
- 4 Оболенский Ю.Г. Управление полетом маневренных самолетов – М. филиал Воениздат, 2007 – 479 с
- 5 Алёшин Б.С., Баженов С.Г., Диденко Ю.И., Шелюхин Ю.Ф. Системы дистанционного управления магистральных самолётов – М. : Наука, 2013. – 292 с.
- 6 Ефремов А.В., Захарченко В.Ф., Овчаренко В.Н. и др. Динамика полётов – М. : Машиностроение-Полет, 2017. – 776 с.
- 7 Овчаренко В.Н. Идентификация аэродинамических характеристик воздушных судов по полётным данным. – М.: Изд-во МАИ, 2017. – 184 с.

Перечень условных обозначений, сокращений и терминов

АСМ	Адаптивная система моделирования
БЛА	Беспилотный летательный аппарат
ВВП	Взлётно-посадочная полоса
ЛА	Летательный аппарат
ЛР	Лабораторная работа
ИУ1	Кафедра «Компьютерные системы и сети» факультета «Информатика и системы управления» МГТУ им Н.Э. Баумана
ОПКС	Собственные общепрофессиональные компетенции
ОС	Операционная система
ПКС	Собственные профессиональные компетенции
ПО	Программное обеспечение
ПОЯ	Предметно-ориентированный язык
ППМ	Поворотные пункты маршрута
СК	Система координат
СУОС	Самостоятельно установленный образовательный стандарт
УК	Управляющий комплекс
ЯП	Язык программирования
SIMODO stars	Прототип адаптивной системы моделирования, разрабатываемой в МГТУ им. Н.Э. Баумана