

# Создание системы управления и анимационное тестирование в реальном времени БЛА самолетных схем

В.В. Макаров, А.В. Полтавский  
ИПУ РАН, г. Москва  
makarov@ipu.ru

*Аннотация.* В работе рассматривается подход к проектированию системы управления беспилотными летательными аппаратами (БЛА). Под БЛА понимаем крылатые аппараты самолетных схем ближнего и среднего радиуса действия. Рассматривается главным образом проектирование программного обеспечения, основанное на использовании моделей и ООП. Такая технология известна как разработка, управляемая моделями (Model Driven Development), и процесс проектирования, управляемый моделями (Model Driven Architecture), с использованием универсального языка моделирования и визуального проектирования UML 2.2. Отраслевые аналитические отчеты показывают значительные преимущества управляемой моделями разработки, а положения о моделировании, добавленные в стандарт DO-178C (ARP4754A), ещё больше повышают важность моделирования в процессе разработки.

Функции пилота ЛА выполняет человек-оператор, включенный в контур управления комплексом. В системе предусмотрена возможность автономного движения БЛА через заданные точки. В данной работе полетное задание определяется как множество точек, заданных в горизонтальной и вертикальной плоскостях, окрестности которых должен пересечь БЛА. Задача состоит в оценке возможности конкретного БЛА (математической модели) пересечь заданные точки или их окрестности и оценить возможное перерегулирование (выбросы траектории при совершении маневра). Проверить, является ли точка выбросом, нетривиальная задача, даже если осуществлять такую оценку на момент начала движения в её сторону. Как минимум на факт достижения такой точки будут влиять масса аппарата, скорость, тяга, расстояние и заданная погрешность автопилота.

Решение задачи, как в большинстве задач анализа движения БЛА, реализуется с помощью имитационного моделирования, как наиболее эффективного средства интерпретации реальности.

В работе приводятся результаты анимационного тестирования отечественных и зарубежных БЛА самолетных схем для задачи прохождения заданных точек с заданной точностью.

*Ключевые слова:* БЛА, UML 2.2, C#, MDD/MDA, IBM Rational Rhapsody, математическая модель движения БЛА, тестирование и анимационное тестирование.

# **Design of UAV control system and animation testing in real time an air bill of aircraft-type UAV**

Vadim Makarov,  
Alexander Poltavsky  
ICS RAS, Moscow

The paper considers the approach to the design of control system of UAV, mainly software, based on the use of models and OOP. This technology is known as Model Driven Development/Model Driven Architecture, using universal modelling language and Visual design UML 2.2 [1, 2]. Industry analysis reports show significant benefits of model driven development and the modeling added to standard DO-178C (ARP4754A), even more increase the importance of modeling in the development process.

In this paper the flight task is defined as a set of points specified in the horizontal and vertical planes, which must cross the UAV. The problem is to estimate the feasibility of different UAVs (mathematical models) cross the specified point or their neighborhood and estimate possible overcontrol (emissions trajectory in-flight maneuver). The pilot actions of aircraft performs an operator, is included in the control loop. It is possible off-line UAV movement through specified points. Check is the point of emission not trivial, even if this estimation execute at the beginning of the movement in the point direction. At least the fact of reaching this point will affect the weight, speed, power engine, distance and accuracy of autopilot system.

The solution of the problem, as most motion analysis problems UAV, is implemented using simulation, as the most effective means of interpreting reality.

The paper cites a number of animation test results UAV for the task of passing points.

The paper presents the results of animated testing of Russian and World UAV aircraft schemes for the problem of passing the specified points with a given accuracy.

## **Введение. Классификация БЛА**

В настоящее время существует потребность в разработке БЛА, способных выполнять различные задачи:

- а) ведение поисковых работ;
- б) аэрофотосъёмка местности;
- в) мониторинг территорий и объектов;
- г) ведение видеонаблюдения и т.д.

Согласно находящимся в открытом доступе документам организаций Европейского Союза, распределение потребительского спроса на гражданские БЛА в период с 2015 по 2020 гг. выглядит следующим образом:

- 45% – правительственные структуры,
- 25% – пожарные,
- 13% – сельское хозяйство и лесничество,
- 10% – энергетика,
- 6% – обзор земной поверхности,
- 1% – связь и вещание [2].

Важнейшая проблема при разработке БЛА - проектирование и разработка системы управления, при этом речь идёт о распределённой системе.

По виду управления БЛА различают:

- а) автономно пилотируемые БЛА (АПЛА);
- б) дистанционно пилотируемые БЛА (ДПЛА).

По типу систем управления для робототехнических систем различают следующие системы управления:

- а) программные;
- б) адаптивные;
- в) интеллектуальные.

В программных (программируемых) системах управления оператором пилотируемого устройства осуществляется программирование, после чего робототехническая система действует, полностью повторяя заданную программу.

В адаптивных системах управления оператором реализуются только основы программы, а БЛА имеет возможность в определенных границах автоматически перепрограммироваться (адаптироваться) в ходе решения поставленной задачи в зависимости от условий, которые неточно определены заранее.

В интеллектуальных системах управления оператором ставится задание в общей форме. Ниже в Табл.1 приводится классификация БЛА [1].

Таблица 1 - Классификация БЛА

БЛА		Радиус действия (РД), км	Высота полета, м	Время полета, час
Тактические	Микро	<10	250	<1
	Мини	<10	500	<2
	Ограниченного РД	10-30	3000	2-4
	Ближнего РД	30-70	3000	3-6
	Среднего РД	70-200	3000-5000	6-10
	Среднего РД с большой продолжительностью полета	500	5000-8000	10-18
	Низковысотные БЛА с большой продолжительностью полета	>500	250-3000	>24
	Низковысотные БЛА дальнего действия	>250	50-5000	0.5-1
	Средневысотные БЛА с большой продолжительностью полета	500-750	5000-8000	24-48
Стратегические	Высотные БЛА с большой продолжительностью полета	1000-6000	15000-20000	24-48
	Стратосферные БЛА	>2000	>20000	>48

Актуальность работы заключается в исследовании современной технологии создания и тестирования СУ БЛА (главным образом программного обеспечения этой распределенной системы), управляемой моделями этой системы и основанной на использовании моделей и ООП. Такая технология известна как разработка, управляемая моделями (Model Driven Development), и процесс проектирования, управляемый моделями (Model Driven Architecture), с использованием универсального языка моделирования и визуального проектирования UML 2.2 [3, 4]. Отраслевые аналитические отчеты показывают значительные преимущества управляемой моделями разработки, а положения о моделировании, добавленные

в стандарт DO-178C (ARP4754A), ещё больше повышают важность моделирования в процессе разработки.

Преимуществами такого подхода согласно являются:

- а) Использование средств для визуального проектирования (UML, SysML, Simulink и т.д.).
- б) Возможность имитации работы системы для проверки разрабатываемого ПО.
- в) Генерирование исходного кода на целевой контроллер.
- г) Тестирование, основанное на моделях, т.е. возможность отслеживать работу системы при помощи анимированных UML-диаграмм.
- д) Возможность отслеживания артефактов жизненного цикла (например, требования, тесты, техническое задание, описание архитектуры и т.д.).

Известны национальные CASE-средства разработки по данному подходу в авиакосмической отрасли. Язык ДРАКОН. Российское космическое агентство (научно-производственный центр автоматики и приборостроения, г. Москва) и Российская академия наук (Институт прикладной математики им. М.В. Келдыша, г. Москва) - проект «Буран».

Система «ГРАФИТ-ФЛОКС». Успешно используется в ряде крупных космических проектов: «Морской старт», «Фрегат», «Протон-М» и др.

### Управление БЛА с помощью команд оператора

Ручное управление реализовано следующим образом. Доступность ручного управления определяется по текущему состоянию БЛА. На текущий момент существуют такие состояния (рис.1):

- а) предстартовое состояние (Idle), соответствующее ожиданию на взлётно-посадочной полосе;
- б) набор высоты (Flight);
- в) ручное управление (Control);
- г) автопилот (MotionInPath);
- д) возвращение на базу (ToGround).

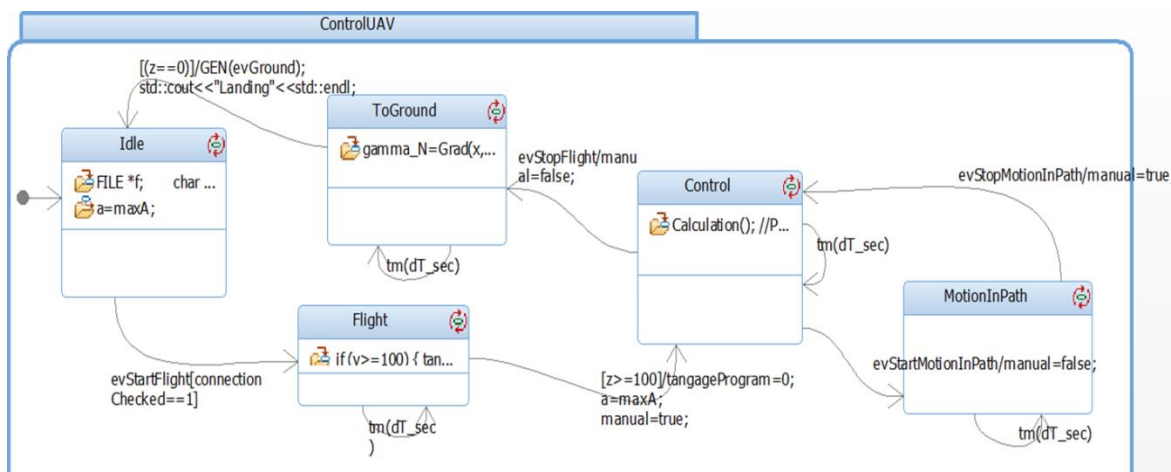


Рисунок 1.- UML-диаграмма состояний БЛА

Основными условиями для перехода между состояниями являются события. Как видно из (рисунка 3.1), условием для перехода между состояниями Idle и Flight является срабатывание события evStartFlight (событие – начало полета). При этом, у этого события есть предикат-защитник connectionChecked, который определяет, что начать взлет нельзя до тех пор, пока не будет успешно проверена связь между наземной станцией и аппаратом.

Условием перехода между состоянием Flight и Control (рисунка 3.1) является достижение аппаратом высоты, большей либо равно 100 метрам над поверхностью земли. При этом целевое значение тангажа (tangageProgram), то есть значение, к которому стремится аппарат,

становится равно 0, что значит, что аппарат начинает выравниваться. Также сила тяги выставляется на максимальное значение, а предикат-защитник активации ручного управления (manual) принимает значение true, что значит, что у оператора стало доступным ручное управление БЛА.

Условием перехода между состоянием Control и MotionInPath является срабатывание события evStartMotionInPath (событие – начало автоматического движения), которое активируется нажатием оператором соответствующей кнопки на панели интерфейса. При этом значение предиката-защитника активации ручного управления (manual) принимает значение false, что отключает возможность ручного управления аппаратом. В свою очередь, срабатывание события evStopMotionInPath (событие – прекращение автоматического движения) возвращает систему к состоянию Control и вновь активирует ручное управление аппаратом.

Условием для перехода между состоянием Control и ToGround является срабатывание события evStopFlight (событие – прекращение полета), которое также активируется нажатием оператором соответствующей кнопки на панели интерфейса. Как и в предыдущем случае, аппарат переходит в режим автоматического движения (возвращение на базу), при котором возможность ручного управления аппаратом отсутствует. Условием для перехода между состоянием ToGround и Idle является успешная посадка аппарата на базу.

### Математическая модель движения БЛА

Для реализации динамики полёта в имитационной модели системы управления БЛА за основу была выбрана модель движения ЛА [1]. Модель движения ЛА представляет собой систему дифференциальных уравнений, описывающих движение в вертикальной и горизонтальной плоскости:

$$\left\{ \begin{array}{l} m\dot{V} = P \cos \alpha - X - mg \sin \theta \\ X = C_{xa} q S_x \\ q = \frac{\rho V^2}{2} \\ mV\dot{\theta} = P \sin \alpha + Y - mg \cos \theta \\ Y = C_y q S_x \\ mV\dot{\varphi} = P \sin \beta + Z \\ P \leq P_{\max} \\ \theta \leq \theta_{\max} \\ Z = C_z q S_x \\ \dot{z} = V \sin \theta \\ \dot{y} = V \sin \varphi \cos \theta \\ \dot{x} = V \cos \theta \cos \varphi \\ \dot{m} = -\mu \\ \delta_B = K_{11}(\vartheta - \vartheta_{\text{пр}}) + K_{12}\dot{\vartheta} \\ \delta_H = K_{21}(\gamma - \gamma_{\text{пр}}) + K_{22}\dot{\gamma} \\ \delta_B \leq \delta_{B \max} \\ \delta_H \leq \delta_{H \max} \\ \vartheta = \theta + \alpha \\ \gamma = \varphi + \beta \end{array} \right.$$

$\gamma_{\text{пр}}$  – заданный угол рыскания;  
 $K_{21}$  – коэффициент руля направления №1;  
 $K_{22}$  – коэффициент руля направления №2  
 $\vartheta_{\text{пр}}$  – заданный угол тангажа;  
 $K_{11}$  – коэффициент руля высоты №1;  
 $K_{12}$  – коэффициент руля высоты №2;  
 $\delta_H$  – отклонение руля направления;  
 $\gamma$  – угол рыскания;  
 $\vartheta_{\text{пр}}$  – заданный угол тангажа;  
 $x$  – координата по оси X;  
 $y$  – координата по оси Y;  
 $z$  – координата по оси Z;  
 $\delta_B$  – отклонение руля высоты;  
 $\vartheta$  – угол тангажа;  
 $\beta$  – угол скольжения;  
 $\varphi$  – угол направления;  
 $Z$  – аэродинамическая боковая сила;  
 $C_z$  – безразмерный аэродинамический коэффициент боковой силы;  
 $P$  – сила тяги двигателя;  
 $m$  – масса летательного аппарата;  
 $V$  – скорость летательного аппарата;  
 $\alpha$  – угол атаки;  
 $X$  – сила лобового сопротивления;  
 $C_{xa}$  – безразмерный аэродинамический коэффициент силы лобового сопротивления;  
 $q$  – скоростной напор;

$C_y$  – безразмерный аэродинамический коэффициент подъемной силы;  
 $\theta$  – угол наклона траектории;  
 $g$  – ускорение свободного падения;

$\dot{H}$  – вертикальная компонента скорости;

$\dot{L}$  – горизонтальная компонента скорости;

$S_x$  – характерная площадь;

$\rho$  – плотность воздуха;

$Y$  – аэродинамическая подъёмная сила;

$\mu$  – количество топлива(кг), сжигаемого в секунду (секундный массовый расход топлива).

Для расчёта угла атаки решается дифференциальное уравнение второго порядка:

$$\ddot{\alpha} + 1,2\dot{\alpha} + 0,3\alpha + 0,3\delta_B = 0$$

Далее это уравнение сводится к системе дифференциальных уравнений первого порядка (путём простой замены).

$$\begin{cases} \alpha = \alpha_1 \\ \dot{\alpha}_1 = \alpha_2 \\ \dot{\alpha}_2 = -1,2\alpha_2 - 0,3\alpha_1 - 0,3\delta_B \end{cases}$$

В большинстве случаев на борту летательного аппарата для решения используется метод Эйлера в связи с необходимостью наискорейшего получения ответа. Применив численный метод Эйлера, получаем:

$$\begin{cases} \alpha_1(t_{m+1}) = \alpha_1(t_m) + dt\alpha_2(t_m) \\ \alpha_2(t_{m+1}) = \alpha_2(t_m) + dt(-1,2\alpha_2(t_m) - 0,3\alpha_1(t_m) - 0,3\delta_B) \end{cases}$$

Аналогичным образом для расчёта угла скольжения решается дифференциальное уравнение второго порядка:

$$\ddot{\beta} + 1,2\dot{\beta} + 0,3\beta + 0,3\delta_H = 0$$

Далее уравнение сводится к системе дифференциальных уравнений первого порядка путём простой замены.

$$\begin{cases} \beta = \beta_1 \\ \dot{\beta}_1 = \beta_2 \\ \dot{\beta}_2 = -1,2\beta_2 - 0,3\beta_1 - 0,3\delta_H \end{cases}$$

Применив численный метод Эйлера, получаем:

$$\begin{cases} \beta_1(t_{m+1}) = \beta_1(t_m) + dt\beta_2(t_m) \\ \beta_2(t_{m+1}) = \beta_2(t_m) + dt(-1,2\beta_2(t_m) - 0,3\beta_1(t_m) - 0,3\delta_H) \end{cases}$$

Коэффициенты подъёмной, боковой аэродинамической силы, а также коэффициент лобового сопротивления рассчитываются следующим образом:

$$\begin{cases} C_{xa} = C_{x0} + C_{xi} \\ C_{xi} = C_{x0}\alpha \\ C_y = C_y^\alpha \alpha \\ C_z = C_z^\beta \beta \end{cases}$$

Если сравнивать приведенное выше математическое описание БЛА с моделями из [5], то наши модели оказываются более простыми, но они позволяют получить все интересующие характеристики динамики БЛА. Решение уравнений проводилось в [5] и в описываемом проекте аналогичными методами.

На следующем рисунке (рис.2) приведен БЛА в скоростной системе координат.

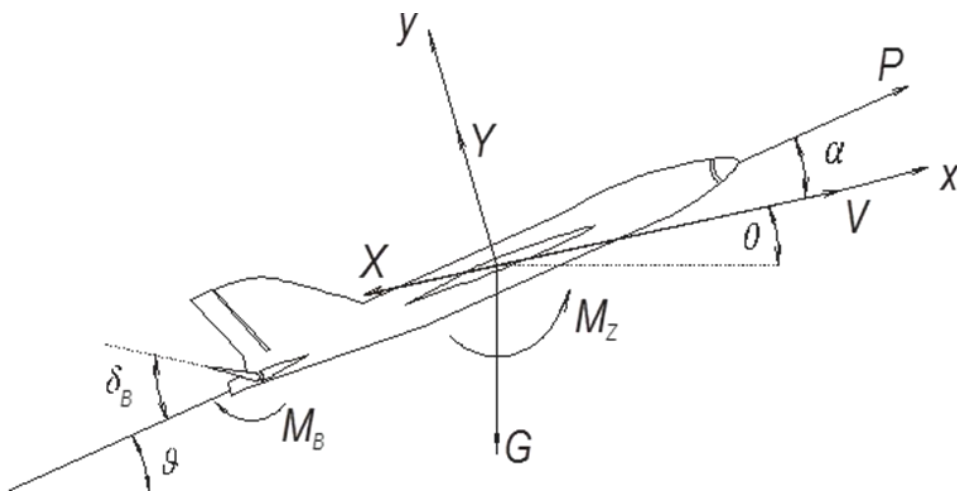


Рисунок 2. БЛА в скоростной системе координат.

### Тестирование и анимационное тестирование разработанной системы

Для загрузки в разрабатываемую систему управления БЛА различных конфигураций летательных аппаратов была разработана подсистема UAVsettings.exe. Данное программное средство позволяет оператору настроить необходимые характеристики аппарата до запуска системы. Благодаря такому подходу значительно упрощаются манипуляции с параметрами и дальнейшие проводимые с системой исследования. В подсистеме также предусмотрена возможность поточечного задания и редактирования маршрута движения, который будет использоваться автопилотом при автоматическом движении БЛА.

Подсистема реализована на языке программирования C#. Для работы данного программного обеспечения с системой её достаточно поместить в папку проекта IBM Rational Rhapsody вместе с XML файлами UAVconfigurations.xml, path.xml, содержащими сохранённые наборы параметров или ЛТХ БЛА и маршруты движения соответственно.

Для инициализации ЛТХ самого БЛА за основу был взят американский стратегический разведывательный БЛА RQ-4 Global Hawk. ЛТХ взяты из [6] (рис.3).

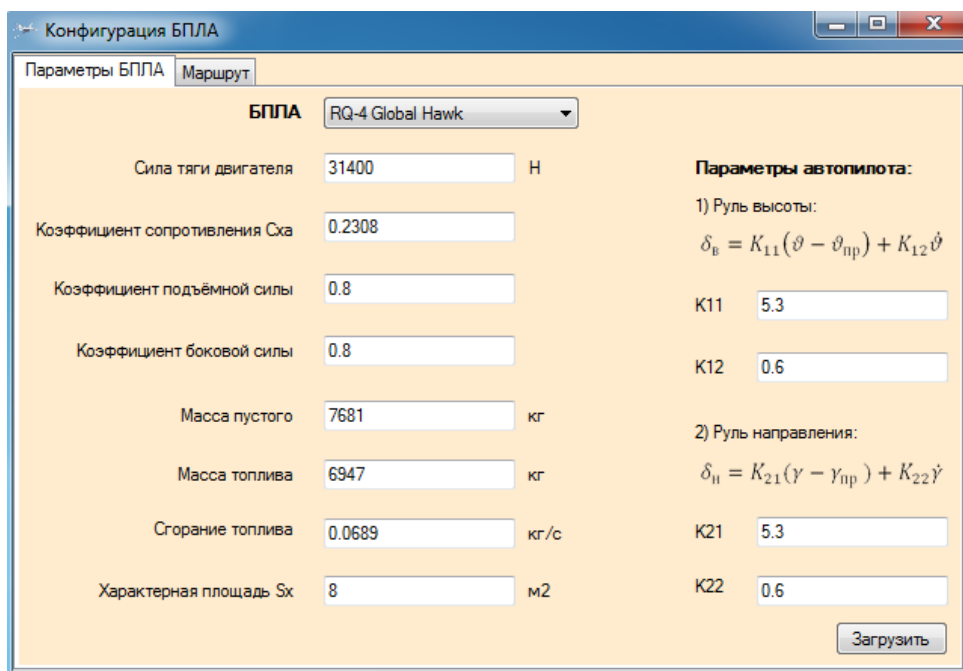


Рисунок 3. Конфигурация БПЛА RQ-4 Global Hawk в UAVsettings.exe

В качестве проверки разработанной системы были проведены тесты для оценки скорости вывода летательного аппарата RQ-4 Global Hawk на высоту 100 м с момента начала разгона по взлётной полосе в зависимости от различной массы топлива на старте.

Таблица 2. Набор высоты в зависимости от массы топлива

№ теста	Масса топлива, кг	Время набора высоты 100 м, сек
1	300	50
2	900	52
3	1500	55
4	2100	57
5	2700	60
6	3300	64
7	3900	66
8	4500	69

Для анализа динамики движения БЛА к анимационной системе тестирования системы управления была добавлена система визуализации положения БЛА в двух плоскостях. На рисунке 4 представлена схема интеграции системы KST и пользовательского приложения. На этой схеме можно выделить три основных объекта:

- Система визуализации.
- Пользовательское приложение.
- Файл с данными.

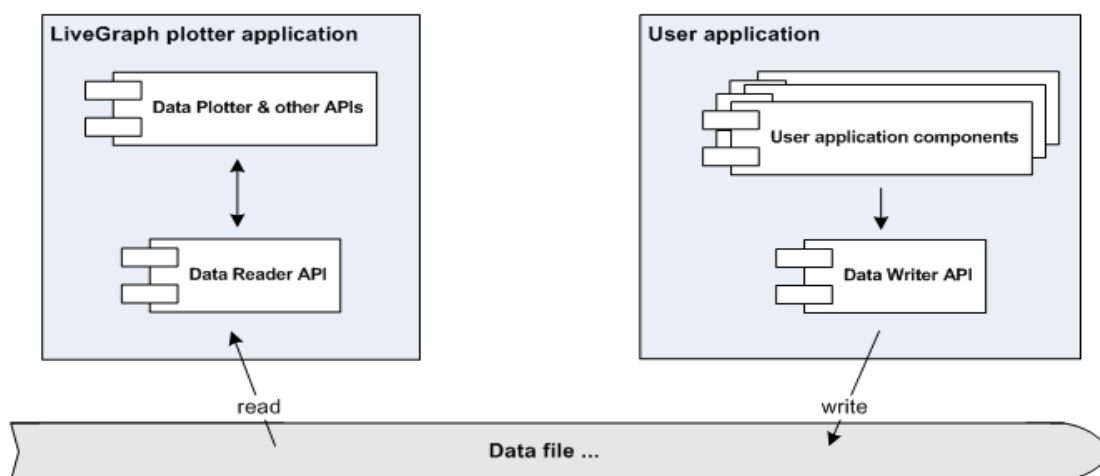


Рисунок 4. Общая схема интеграции системы визуализации и пользовательского приложения

В общем случае, пользовательское приложение, используя API для записи данных, записывает все поступившие от приложения данные в файл, после чего система визуализации, при помощи считывающего API, открывает данный файл, считывает его содержимое и визуализирует его содержимое. Для «поточковых» данных записывающий API хранит выходной поток открытым и записывает новые данные в файл всякий раз, когда приложение передает ему новый набор данных. В тоже время, система визуализации хранит входной поток, на том же самом файле, открытым. С заданным интервалом система визуализации проверяет этот файл на наличие изменений с момента последней проверки. Если файл данных был изменен, система визуализации считывает данные, добавленные после последнего обновления и обновляет график.



В проекте место записывающего API занимает процедура `sendParameterTwo()`, а система визуализации – Kst. Первоначально, в проекте присутствовал только процедура `sendParameter()`, который записывает в файлы телеметрии (текстовые файлы формата .txt) только один входной параметр. Для задач визуализации этого было недостаточно, так как было необходимо визуализировать движение БЛА на плоскостях XY и XZ. Для этого был создан новый процедуры под названием `sendParameterTwo()`, которые является модификацией процедуры `sendParameter`.

Также были проведены тесты для оценки выбросов при движении БЛА по заданному маршруту в горизонтальной плоскости. Под выбросом понимается такая точка заданного маршрута, которая оказывается недостижимой для автопилота системы управления, то есть при движении к такой точке аппарат просто не успевает повернуть так, чтобы попасть в заданную точку. Проверить, является ли точка выбросом, нетривиальная задача, даже если осуществлять такую оценку на момент начала движения аппарата в её сторону. Как минимум на факт достижения такой точки будут влиять масса аппарата, скорость, тяга, расстояния X, Y и заданная погрешность для автопилота  $\epsilon$ . При этом масса летательного аппарата постоянно меняется вследствие выгорания топлива. На рисунке 5 показан алгоритм движения к следующей точке маршрута.

Тесты проводились с заданной тягой двигателя и погрешностью автопилота. Цель: оценить соотношение расстояний X, Y, при котором точка маршрута становится выбросом. Тяга двигателя устанавливается равной половине от максимальной (для RQ-4 Global Hawk это 15700 Н),  $\epsilon = 300$  м.

Проведённые тесты показывают, что летательному аппарату для того, чтобы отклониться от маршрута на 1000 метров нужно приблизительно 3000 метров. Получили соотношение приблизительно 3:1. Видно также, что с ростом массы БЛА это соотношение меняется. На каждые 600 кг дополнительной массы аппарату требуется приблизительно на 250 метров больше для успешного отклонения на 1 км.

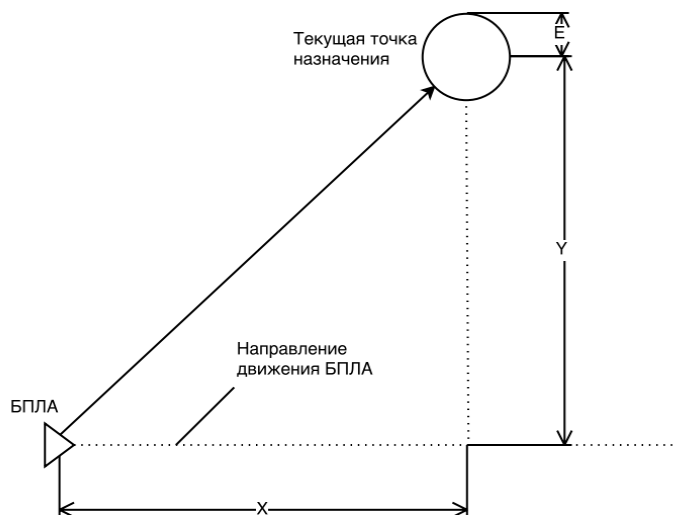


Рисунок 5. Движение БЛА к следующей точке маршрута.

Таблица 3 - Возможность достижения точки аппаратом в зависимости от соотношения X:Y и его массы

№ теста	Масса топлива, кг	X, м	Y, м	Точка достигнута, да/нет
1	300	1000	1000	нет
2		3000		да

№ теста	Масса топлива, кг	X, м	Y, м	Точка достигнута, да/нет
3		2000		нет
4		2500		нет
5		2750		да
6		2700		нет
7	900	2750	1000	нет
8		3000		да
9	1500	3000	1000	нет
10		3250		да

На рисунке 6 показан многооконный интерфейс анимационного тестирования СУ БЛА. Слева на черном экране отражается в динамике телеметрия с БЛА и команды к нему. Далее слева направо – интерфейс системы разработки при анимационном тестировании. Следующий – основной интерфейс оператора БЛА. Два вспомогательный интерфейса оператора - загрузка параметров БЛА и маршрута (контрольный точек). Последние два интерфейса отражают движение БЛА в двух плоскостях. Для более удобного анализа процесса функционирования системы интерфейсы располагаются на нескольких мониторах (в данной работе на двух из-за недостаточного финансирования).

Зададим маршрут и протестируем его прохождение различными летательными аппаратами. Координаты маршрута, заданные в разработанной подсистеме UAVsettings.exe приводятся на рисунке ниже. ЛТХ беспилотных летательных аппаратов из [2, 5, 6].

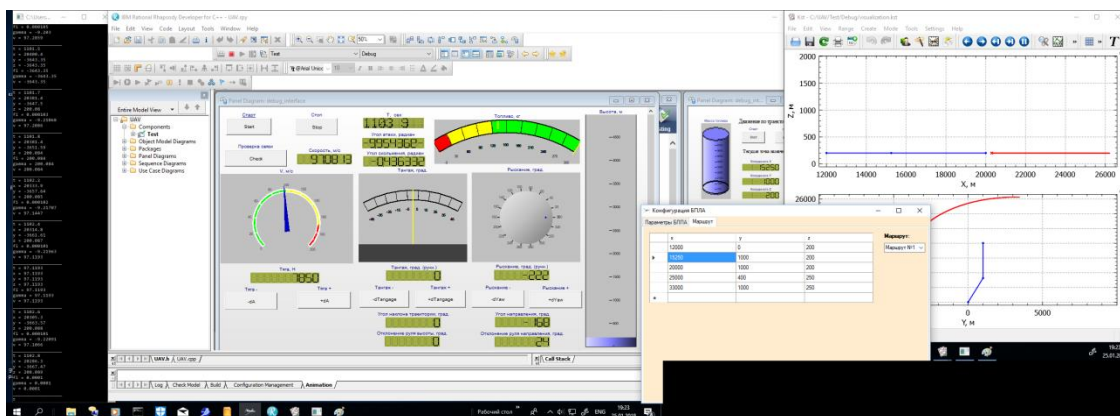


Рисунок 6. Многооконный интерфейс оператора (движение БПЛА RQ-4 Global Hawk через заданные опорные точки)

Таблица 4 - Результаты прохождения маршрута БЛА X-47A при различных уровнях начальной заправки

№ теста	Масса топлива, кг	Разница между планируемым маршрутом и пройденной траекторией, м
1	300	28,6
2	800	36,3
3	1238	47,4

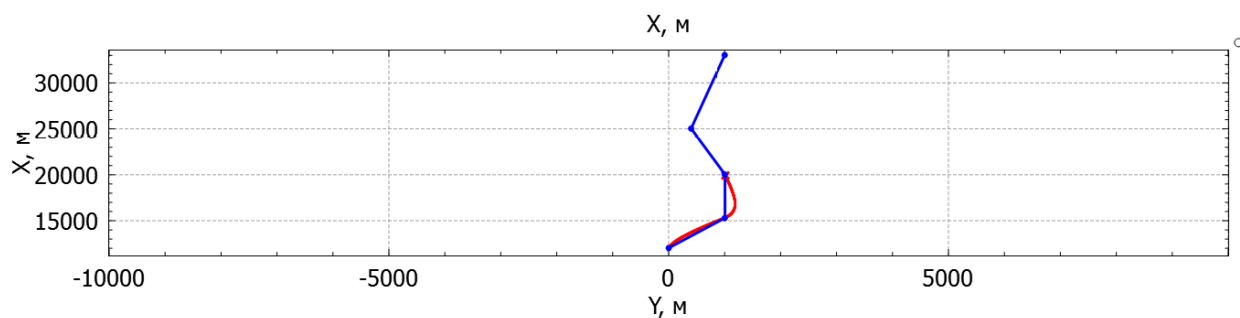


Рисунок 7. - Визуализация движения БЛА X-47A по маршруту в горизонтальной плоскости (Тест №3)

В заключении следует отметить, что в фундаментальном труде «Концерн «Вега» [5] в качестве среды исследования/разработки использовался Matlab/Simulink, более дорогой инструмент и менее эффективный. Судя по публикации, авторы не смогли реализовать систему разработки для различных БЛА. Однако обоснование этого утверждения не является предметом данной работы.

### Литература

1. Полтавский, А.В. Многофункциональные комплексы беспилотных летательных аппаратов [Текст]: научное издание / А.В. Полтавский, А.А. Бурба, А.Е. Аверкин, В.В. Макаров, В.В. Маклаков. - М.: ИПУ РАН, 2015. - 127 с.
2. Иванов М.С. Беспилотные летательные аппараты. [Текст]: Справочное пособие. / под общ. ред. С.А. Попова; ВУНЦ ВВС «ВВА». – Воронеж: Издательско-полиграфический центр «Научная книга», 2015. 619с.
3. Douglass V. P. Real-Time UML Workshop for Embedded Systems. [Текст]: – Oxford, UK, Elsevier Inc., 2007. p.408.
4. Зыль С.Н. Проектирование, разработка и анализ программного обеспечения систем реального времени. – СПб.: БХВ-Петербург, 2010. 336с.
5. Комплексы с беспилотными летательными аппаратами. [Текст]: В 2-х кн.: Кн.1. Принципы построения и особенности применения комплексов с БЛА. Монография/ под ред. В.С. Вербы, Б.Г. Татарского. – М.: Радиотехника, 2016. – 512 с. Кн. 2. Робототехнические комплексы на основе БЛА. Монография/ под ред. В.С. Вербы, Б.Г. Татарского. – М.: Радиотехника, 2016. 824 с.
6. Разведывательный БПЛА RQ-4 Global Hawk // Беспилотные летательные аппараты URL: <http://bp-la.ru/strategicheskij-razvedyvatelnyj-bpla-rq-4-global-hawk/> (дата обращения: 08.04.2017).