Санкт-Петербургский государственный университет
Факультет прикладной математики − процессов управления
Кафедра механики управляемого движения

Цуканов Александр Валерьевич

Выпускная квалификационная работа бакалавра

Разработка программного комплекса построения оптимальной траектории движения БПЛА

Направление 010300
Фундаментальная информатика и информационные технологии

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Шиманчук Д.В.

Санкт-Петербург
2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В последнее время в связи с бурным развитием техники и электроники большой интерес исследователей привлекает создание и управление беспилотными летательными аппаратами. **Беспилотный летательный аппарат** или **БПЛА** (в СМИ упоминается, как «**дрон**» или «**беспилотник**») − летательный аппарат без человеческого экипажа на борту, который осуществляет полёт с помощью удаленного управления с поверхности Земли или по предварительно запрограммированным траекториям. Повышение интереса к разработке программного обеспечения для БПЛА связано с тем, что они обладают некоторыми преимуществами в сравнении с традиционными пилотируемыми летательными аппаратами. **В числе достоинств** БПЛА можно назвать их успешную применимость в задачах, являющихся слишком сложными, опасными, или же монотонными для обычных летательных аппаратов с экипажем на борту. Данное направление исследований является актуальным, так как летательные аппараты такого типа находят широкое применение как в гражданских, так и военных задачах, таких как: аэронаблюдение, видеосъёмка и фотография, а также разведка, поисково-спасательные операции, мониторинг окружающей среды, различные задачи служб пожарной безопасности и сельского хозяйства.

На сегодняшний день используются различные способы управления БПЛА: полёт может управляться разными способами с вариативной автономией − например, управляться оператором с поверхности Земли или с подвижного объекта (автомобиля и т.п.) или управляться полностью автономно с помощью встроенного компьютера и датчиков.

Для управления дроном необходимо иметь набор сенсоров, которые могли бы собирать информацию и определять параметры, характеризующие движение БПЛА. В их число входят:

- Гироскоп (для определения углов ориентации)

- GPS или ГЛОНАСС модуль (для определения положения и скорости)
- Контроллеры скорости (для определения скорости)
- Акселерометр (для определения перегрузок)
- Альтиметр (для определения высоты)
- Компас (для определения направления относительно севера)
- Радар (для определения окружающих объектов)
- Камера (для определения местоположения)

Большинство БПЛА используют радиочастотные приёмники, которые соединяют антенну с цифро-аналоговым преобразователем, и бортовой компьютер, управляющий авионикой (который может управлять летательным аппаратом полуавтономно или автономно). Радиопередатчик используется для приёма команд управления дроном и для передачи всевозможных данных (например видео или фото), телеметрии и статуса своих систем.

В качестве управляющей аппаратуры (если речь идёт о полностью автономных дронах небольшого размера), как правило, используются системы с архитектурой RISC (сокращённый набор команд), однокристальные системы (SOC) или вычислители на базе SBC (одноплатный компьютер) − все они позволяют достигнуть максимального быстродействия, сохраняя малый вес аппаратуры. Наоборот же, в больших БПЛА могут использоваться полноценные компьютеры, так как их грузоподъёмность не побуждает инженеров заниматься миниатюризацией аппаратного обеспечения.

Классические операционные системы не подходят для управления БПЛА, так как долгие временные задержки могут оказаться фатальными для летательного аппарата. ОС для дронов могут быть дополненными ПО RaspberryPis, Beagleboards или быть разработаны как самостоятельные, чтобы подходить под жёсткие требования для работы в режиме реального времени.

Зачастую в БПЛА встраивают программное обеспечение, реализующее автопилотирование дроном, упрощающее управление аппаратом. Автопилот полностью берёт на себя выбор траектории, регулирование тяги двигателя и рулей. Данный способ управления является предпочтительным, так как исключает влияние человеческого фактора, а значит и сопряженные с ним ошибки, а также позволяет избежать крушений БПЛА, происходящих по причине потери связи с центром управления. Это приводит к необходимости создания алгоритмов, реализующих автоматическое построение траектории, переводящей БПЛА из одной точки в другую. Что немало важно построенная траектория должна быть оптимальной, так как минимизация расхода топлива, а также времени полета являются одними из наиболее важных условий эффективного использования БПЛА. Следовательно, существует потребность в реализации точного алгоритма, способного предлагать оптимальную траекторию, не затрачивая на это много времени.

ПОСТАНОВКА ЗАДАЧИ

В данной работе будет рассмотрен полностью автономный вариант управления БПЛА. Основной проблемой данной работы является построение компьютерной модели БПЛА, которая была бы способна предложить оптимальную траекторию полёта беспилотника (время манёвра по которой было бы минимальным), приводящую летательный аппарат из некоторого заданного начального состояния в заданное конечное состояние.

Поставленную задачу и её решение можно разбить на следующие подзадачи:

- Описание параметров физической модели БПЛА
- Описание математической модели движения БПЛА
- Решение задачи терминального управления, обеспечивающее получения управлений, которые бы могли реализовать переход БПЛА из начального состояния в конечное с требуемыми ограничениями.
- Разработка алгоритма для поиска оптимального времени полёта (Δt) при ограничении на переменные состояния и управления.
- Нахождение соответствующей оптимальному промежутку времени Δt программной траектории и управлений.
- Создание программного комплекса, предполагающего визуализацию результатов

ОБЗОР ЛИТЕРАТУРЫ

Благодаря бурному развитию в области управления беспилотными летательными аппаратами, за последние годы написано множество монографий, содержащих основные понятия данного направления исследований.

Одним из учёных, положивших начало в этой области исследований является П. Крутько. Именно материал из его книги «Обратная задача динамики управляемых систем» [1] был взят за основу для разработки математической модели. В его работе содержатся описания движения летательного аппарата в различных плоскостях. В результате им была получена полная математическая модель движения БПЛА, описываемая дифференциальными уравнениями. Так как полная математическая модель оказалась вычислительно сложной, для дальнейших упрощений была использована работа В. Моисеева «Теория управления беспилотными летательными аппаратами» [2]. Благодаря этой книге и книге А. Колесникова «Новые нелинейные методы управления полетом» [3] была получена упрощённая математическая модель, которую и было решено использовать в дальнейшем. Были также рассмотрены работы: Т. Касаткиной [4], в которой содержалась информация о преобразованиях линейных систем к каноническому виду; статья А. Канатникова и Е. Шмагиной «Задача терминального управления движениями летательного аппарата» [5], в которой были описаны задачи управления ЛА и методы решения задачи терминального управления.

Для реализации программного комплекса и имплементации алгоритма поиска оптимальной траектории была использована онлайн документация к вычислительному пакету MATLAB [6].

ГЛАВА 1. МОДЕЛИРОВАНИЕ БПЛА КАК ОБЪЕКТА УПРАВЛЕНИЯ

Для программного построения траектории БПЛА необходимо рассмотреть его математическую модель как объекта для автоматического управления. В данной главе будут представлены физические характеристики летательного аппарата, а также модель управляемого движения БПЛА.

Описание математической модели БПЛА

Выбор систем координат

Для описания положения и движения БПЛА в пространстве могут использоваться различные системы координат. Обычно выбор конкретной системы координат диктуется контекстом поставленной задачи.

1.1. Определение положения летательного аппарата

Стандартным подходом для определения координат БПЛА является описание положения его **центра масс** относительно Земли, которое в данной работе будет определяться в **нормальной земной системе координат** (**НЗСК**). Под **НЗСК** будем понимать правую декартову прямоугольную систему координат Oxyz, которая жёстко связана с Землей и является инерциальной [3]. Начало этой системы координат располагается на земной поверхности в некоторой неподвижной точке O. Ось Oy направлена по вертикали против вектора силы тяжести. В данной модели делается допущение о том, что Земля является плоской (то есть модель пренебрегает кривизной поверхности Земли). Земля в данной системе представляется, как плоскость, образуемая осями координат x и z, при y=0. Следовательно,

положение центра масс ЛА относительно Земли можно описать, задавая координаты (x, y, z). Координата x характеризует дальность полёта БПЛА L, координата z характеризует боковое смещение Z, а координата y описывает высоту полёта H.

1.2. Определение направление движения летательного аппарата

За направление движения летательного аппарата будем принимать ориентацию его вектора скорости V. Для определения направления движения введем в рассмотрение **траекторную систему координат**, которую обозначим как $O_t x_t y_t z_t$ [5]. Данная система координат жёстко связана с центром масс БПЛА, причём её начало O_t есть центр масс летательного аппарата. Ось $O_t x_t$ направлена вдоль земной скорости ЛА, ось $O_t y_t$ перпендикулярна оси $O_t x_t$ и направлена вверх относительно местной вертикали, а ось $O_t z_t$ дополняет предыдущие оси до правой тройки векторов. Следовательно, определить направление движения БПЛА можно путем ориентации осей данной системы координат относительно НЗСК. Положение траекторной системы координат задаётся путевым углом ψ и углом наклона траектории ϑ.

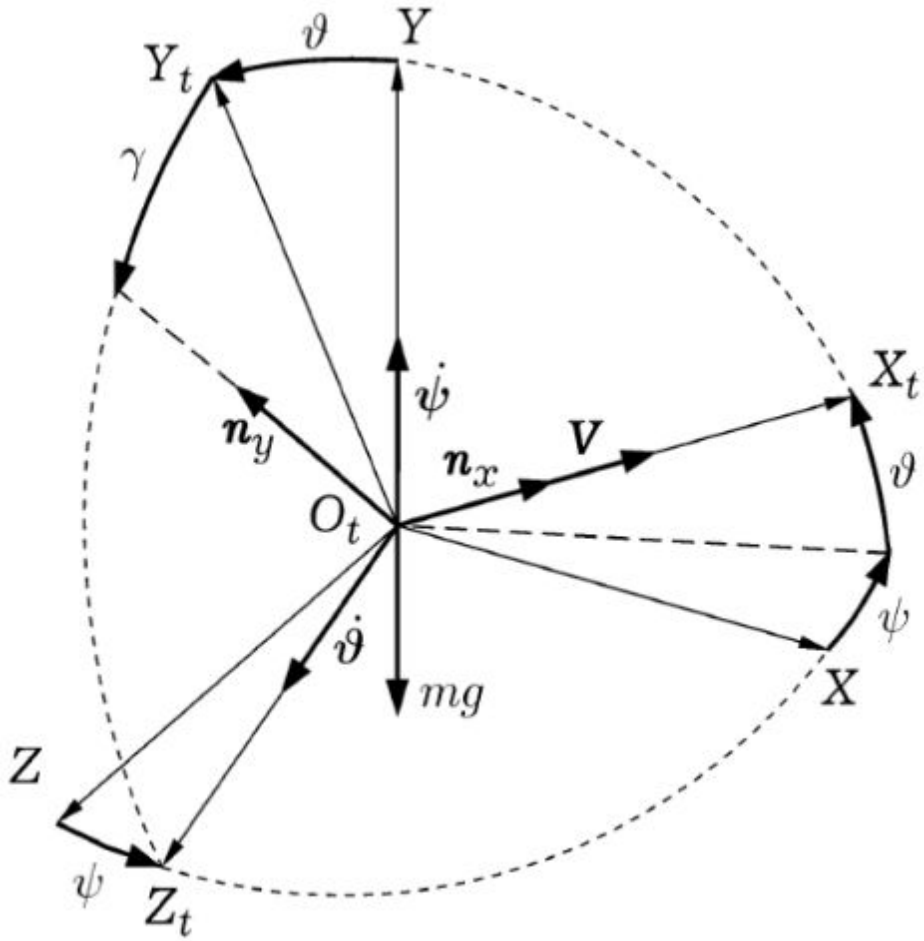

Рис. 1. Нормальная земная система координат и траекторная система координат.
На рисунке показан вектор скорости, углы наклона траектории, курсовой угол, угол крена и перегрузки

1.3. Описание основных параметров физической модели ЛА

Основные параметры физической модели летательного аппарата, которые понадобятся в дальнейшем для описания математической модели:

1. H − высота полёта
2. L − дальность полёта
3. Z − боковое смещение
4. V − путевая скорость
5. ϑ − угол наклона траектории
6. ψ − путевой угол

7. γ − угол крена
8. n_X − продольная перегрузка
9. n_Y − поперечная перегрузка
10. g − ускорение свободного падения

1.4. Описание модели движения БПЛА

В данной работе летательный аппарат рассматривается как свободное твёрдое тело, имеющее шесть степеней свободы.

Рассмотрим **динамические уравнения движения** центра масс БПЛА в траекторной системе координат [1].

$$\begin{cases} m\dot{V} = \sum F_{x_t}, \\ mV\dot{\vartheta} = \sum F_{y_t}, \\ -mV\dot{\psi}\cos(\vartheta) = \sum F_{z_t}. \end{cases} \quad (1.2.1)$$

В рамках данной модели рассмотрим следующие силы, действующие на летательный аппарат в процессе полёта:

1. G = mg − сила тяжести
2. P − сила тяги
3. X` - сила лобового сопротивления
4. Y` − подъёмная сила
5. Z` − боковая сила

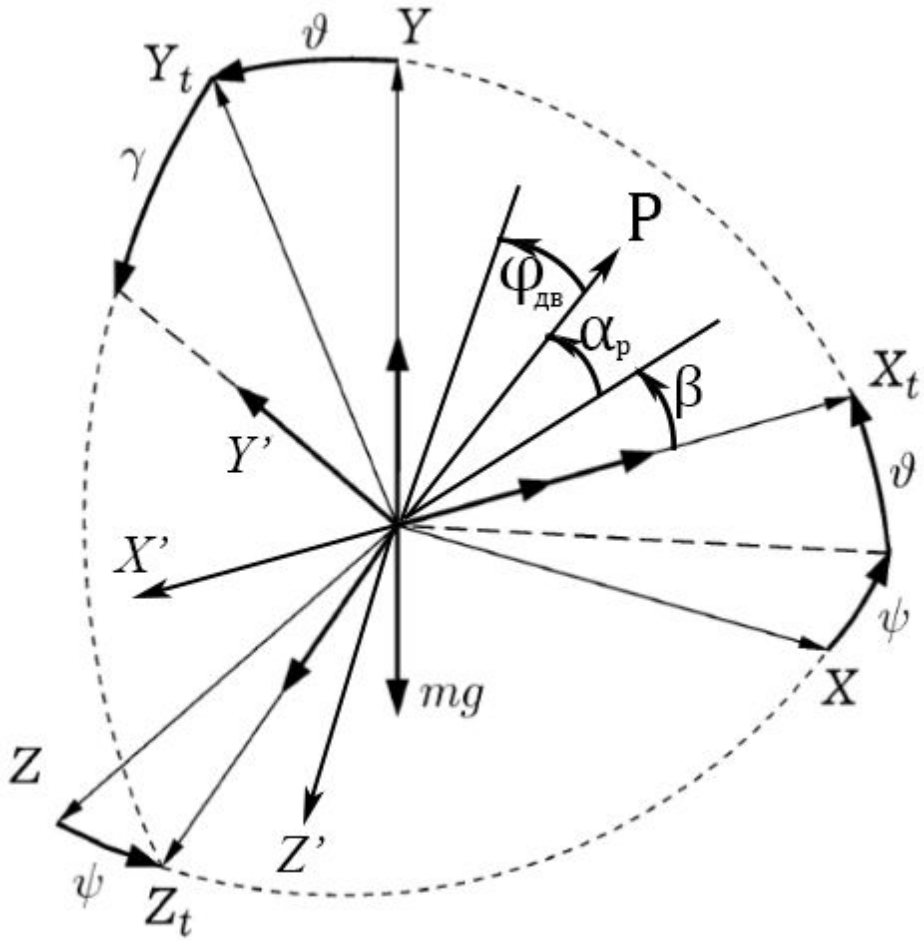

Рис. 2. Нормальная земная и траекторная система координат с отмеченными векторами сил, приложенных к БПЛА

Пользуясь рис. 2, можно получить проекции всех упомянутых выше сил на оси траекторной системы координат. В итоге для уравнений (1.2.1) получаем следующие равенства:

$$\begin{cases} m\dot{V} = P\cos(\alpha_p)\cos(\beta) - X' - mg\sin(\vartheta), \\ mV\dot{\vartheta} = (Y' + P\sin(\alpha_p)\cos(\gamma) - (Z' - P\cos(\alpha_p)\sin(\beta))\sin(\gamma) - mg\cos(\vartheta), \\ -mV\dot{\psi}\cos(\vartheta) = (P\sin(\alpha_p) + Y')\sin(\gamma) + (Z' - P\cos(\alpha_p)\sin(\beta))\cos(\gamma). \end{cases}$$

(1.2.2)

При решении практических расчетных задач, таких как построение траектории, считаются справедливыми следующие приближенные равенства [3]:

$$sin\alpha \approx \alpha,\ sin\beta \approx \beta,\ cos\alpha \approx 1,\ cos\beta \approx 1.$$

Принимая во внимание описанные соотношения, система уравнений (1.2.2) может быть переписана следующим образом:

$$\begin{cases} \dot{V} = \frac{P-X'}{m} - g\sin(\vartheta), \\ \dot{\vartheta} = \frac{(Y'+P\alpha_p)\cos(\gamma)-(Z'-P\beta)\sin(\gamma)}{mV} - \frac{g}{V}\cos(\vartheta), \\ \dot{\psi} = -\frac{(P\alpha_p+Y')\cos(\gamma)+(Z'-P\beta)\sin(\gamma)}{mV\cos(\vartheta)}. \end{cases} \quad (1.2.3)$$

В данной работе рассматривается такой способ бокового маневрирования, при котором необходимая боковая сила возникает за счёт изменения угла крена. В данном случае при построении модели принимают $\beta = 0$, $Z' = 0$ [2]. Подставив данные значения в систему (1.2.3), придём к следующей системе дифференциальных уравнений:

$$\begin{cases} \dot{V} = \frac{P-X'}{m} - g\sin(\vartheta), \\ \dot{\vartheta} = \frac{(Y'+P\alpha_p)\cos(\gamma)}{mV} - \frac{g}{V}\cos(\vartheta), \\ \dot{\psi} = -\frac{(P\alpha_p+Y')\sin(\gamma)}{mV\cos(\vartheta)}. \end{cases} \quad (1.2.4)$$

При рассмотрении движения БПЛА часто используют понятие перегрузки, через которые и выражают динамические уравнения движения. Приближенные компоненты вектора перегрузки выражаются следующим образом [1]:

$$n_x = \frac{P - X'}{mg}, \qquad n_y = \frac{Y' + P\alpha_p}{mg}.$$

Подставив данные выражения в уравнения (1.2.4), перейдем к следующей системе динамических уравнений в перегрузках:

$$\begin{cases} \dot{V} = g(n_x - \sin(\vartheta)), \\ \dot{\vartheta} = g\frac{n_y\cos(\gamma)-\cos(\vartheta)}{V}, \\ \dot{\psi} = -g\frac{n_y\sin(\gamma)}{V\cos(\vartheta)}. \end{cases} \quad (1.2.5)$$

Динамические уравнения вращения БПЛА для упрощения модели рассмотрены не будут. Будем считать, что при рассматриваемом управлении БПЛА время переходных процессов по каналам тангажа, рысканья и крена должны быть малыми, благодаря чему можно сделать предположение о том,

что углы атаки α, скольжения β и крена γ устанавливаются мгновенно при изменении положений соответствующих органов управления БПЛА. Такое предположение об идеальности системы управления БПЛА позволяет считать его **безынерционным вращающимся объектом**, для которого выполняются условия равенства нулю главных центральных моментов инерции, а следовательно оказываются равны нулю и суммы проекций моментов всех действующих на БПЛА сил относительно его центра масс [2].

Возможность практического применения данной модели дополнительно обосновывается тем, что движение центра масс БПЛА обладает большой инерционностью, а его колебания вокруг ЦМ вызывают сравнительно малые отклонения траектории полета БПЛА [2], что вписывается в решение поставленной задачи.

Положение центра масс относительно неподвижной земной системы координат определяется следующими **кинематическими уравнениями [1]**:

$$\begin{cases} \dot{H} = V\sin(\vartheta), \\ \dot{L} = V\cos(\vartheta)\cos(\psi), \\ \dot{Z} = -V\cos(\vartheta)\sin(\psi). \end{cases}$$

Совместив полученные уравнения, получим следующую математическую модель, представляющую собой систему из шести дифференциальных уравнений, описывающих положение и движения центра масс БПЛА [5]:

$$\begin{cases} \dot{H} = V\sin(\vartheta), & \dot{V} = g(n_x - \sin(\vartheta)), \\ \dot{L} = V\cos(\vartheta)\cos(\psi), & \dot{\vartheta} = g\frac{n_y\cos(\gamma)-\cos(\vartheta)}{V}, \\ \dot{Z} = -V\cos(\vartheta)\sin(\psi), & \dot{\psi} = -g\frac{n_y\sin(\gamma)}{V\cos(\vartheta)}. \end{cases}$$

В данной системе уравнений тройка координат (H; L; Z) представляет собой координаты положения центра масс летательного аппарата в нормальной земной неподвижной системе координат (п 1.1.1). Перегрузки n_X,

n_Y и угол крена в данной системе рассматриваются **в качестве управлений** движением ЛА. Углы ϑ и ψ задают направление траектории, а именно – касательную к ней. Соответственно они так же задают направление вектора земной скорости V. На рис. 3 ниже показано, как задаётся вектор земной скорости БПЛА и угол крена летательного аппарата:

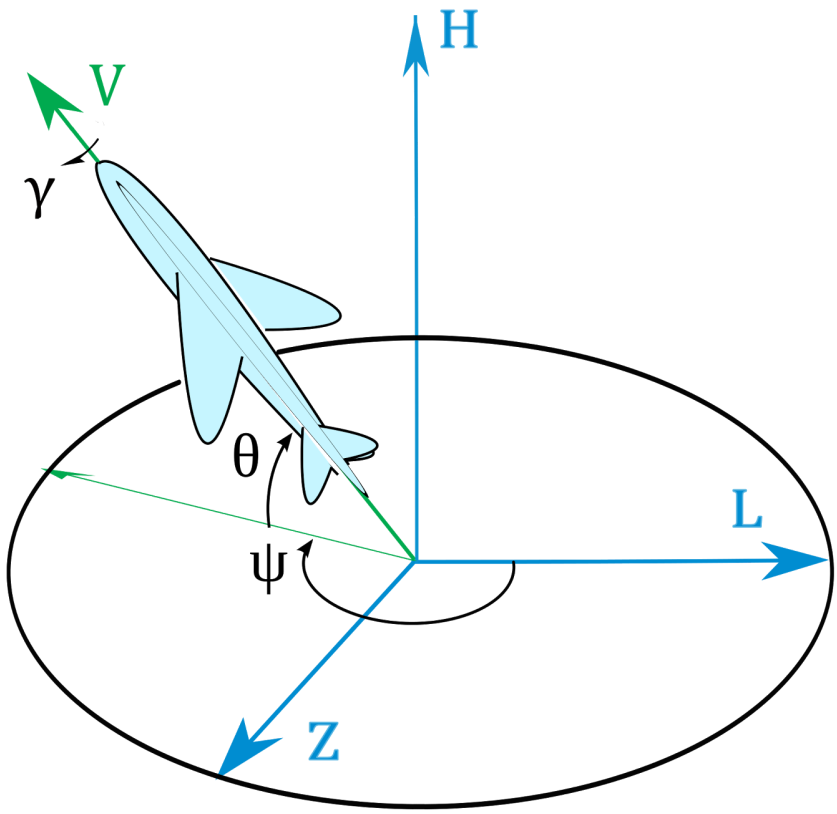

Рис. 3. Параметры, задающие положение траектории БПЛА в пространстве

ГЛАВА 2. ЗАДАЧА ТЕРМИНАЛЬНОГО УПРАВЛЕНИЯ

Рассмотрим задачу терминального управления полётом летательного аппарата, которое приводило бы БПЛА из заданного начального состояния (V_0; H_0; L_0; Z_0; ϑ_0; ψ_0), (n_{X0}; n_{Y0}; γ_0) при t=0 в некоторое конечное состояние (V_*; H_*; L_*; Z_*; ϑ_*; ψ_*), (n_{X*}; n_{Y*}; γ_*) при t=Δt. Допустим, что Δt уже известно заранее. Заданные состояния называются **граничными условиями** задачи терминального управления [5].

2.1 Решение задачи терминального управления

Для того, чтобы получить траекторию, необходимо определить функции L, Z и H, благодаря которым траектория БПЛА и описывается. Для этого воспользуемся обратными методами динамики. Для получения функций управлений введём новые управления по формулам $u_1=n_X$, $u_2=n_Y*\cos(\gamma)$, $u_3=n_Y*\sin(\gamma)$. Тогда уравнения математической модели движения БПЛА являются линейными по новым управлениям:

$$\begin{cases} \dot{H} = V\sin(\vartheta) & \dot{V} = g(-\sin(\vartheta) + u_1) \\ \dot{L} = V\cos(\vartheta)\cos(\psi) & \dot{\vartheta} = g(-\frac{\cos(\vartheta)}{V} + \frac{1}{V}u_2) \\ \dot{Z} = -V\cos(\vartheta)\sin(\psi) & \dot{\psi} = -\frac{g}{V\cos(\vartheta)}u_3 \end{cases} \quad (2.1.1)$$

Далее для получения управлений приведём систему дифференциальных уравнений (2.1.1) к каноническому виду, введя новые переменные состояния [4]:

$x_1=H$, $x_2=L$, $x_3=Z$, $x_4=V*\sin(\vartheta)$, $x_5=V*\cos(\vartheta)*\cos(\psi)$, $x_6= -V*\cos(\vartheta)*\sin(\psi)$.

Заметим, что благодаря данной замене выполняются соотношения $\dot{x}_1 = x_4, \dot{x}_2 = x_5, \dot{x}_3 = x_6$.

Новые переменные состояний с ограничениями $\{|\vartheta|<\pi/2;\ |\psi|<\pi;\ V>0\}$ определяют гладкую невырожденную замену переменных, так как старые переменные состояния могут быть выражены с помощью новых по формулам:

$$\begin{cases} H = x_1, & V = \sqrt{x_4^2 + x_5^2 + x_6^2}, \\ L = x_2, \quad Z = x_3, & \sin(\vartheta) = \frac{x_4}{\sqrt{x_4^2+x_5^2+x_6^2}}, \\ \sin(\psi) = -\frac{x_6}{\sqrt{x_5^2+x_6^2}}, & \cos(\psi) = \frac{x_5}{\sqrt{x_5^2+x_6^2}}. \end{cases}$$

Таким образом, в новых переменных получаем систему дифференциальных уравнений в каноническом виде:

$$\begin{cases} \dot{x}_1 = x_4, \\ \dot{x}_4 = -g + u_1 g \sin(\vartheta) + u_2 g \cos(\vartheta), \\ \dot{x}_2 = x_5, \\ \dot{x}_5 = u_1 g \cos(\vartheta)\cos(\psi) - u_2 g \sin(\vartheta)\cos(\psi) + u_3 g \sin(\psi), \\ \dot{x}_3 = x_6, \\ \dot{x}_6 = -u_1 g \cos(\vartheta)\sin(\psi) + u_2 g \sin(\vartheta)\sin(\psi) + u_3 g \cos(\psi). \end{cases}$$

Выражая x_4, x_5, x_6 через x_1, x_2, x_3, можно получить систему из трёх дифференциальных уравнений, которые разрешимы относительно новых управлений:

$$\begin{cases} u_1 = \frac{(\ddot{x}_1+g)\sin(\vartheta)+\ddot{x}_2\cos(\vartheta)\cos(\psi)-\ddot{x}_3\cos(\vartheta)\sin(\psi)}{g}, \\ u_2 = \frac{(\ddot{x}_1+g)\cos(\vartheta)-\ddot{x}_2\sin(\vartheta)\cos(\psi)+\ddot{x}_3\sin(\vartheta)\sin(\psi)}{g}, \\ u_3 = \frac{\ddot{x}_2\sin(\psi)+\ddot{x}_3\cos(\psi)}{g}. \end{cases}$$

Из полученного выше следует, что для любой траектории, заданной функциями от времени $x_1(t)$, $x_2(t)$, $x_3(t)$, будут найдены реализующие эту траекторию управления. Однако, задача заключается в поиске траектории, удовлетворяющей наложенным на неё граничным условиям, поэтому

необходимо показать, при каких условиях функции $x_1(t)$, $x_2(t)$, $x_3(t)$ будут подходить под граничные условия. Итак, граничные условия на переменные состояния и на управления определяют граничные значения функций $x_1(t)$, $x_2(t)$, $x_3(t)$ на концах отрезка времени $[0; \Delta t]$, а также их первые и вторые производные (граничные условия для новых переменных состояния и управлений легко получить, подставляя изначальные граничные значения задачи в формулы, по которым производилась замена переменных). По сути любые гладкие функции, удовлетворяющие полученным граничным условиям, будут являться решением терминальной задачи.

2.2 Построение пространственной траектории

Ранее было показано (п. 2.1), что для траектории, заданной гладкими функциями $x_1(t)$, $x_2(t)$, $x_3(t)$ и удовлетворяющей условиям, полученных из граничных условий для переменных состояния, обязательно найдутся реализующие её управления. Например, такие функции можно найти среди многочленов пятой степени, в силу того что на каждую функцию накладывается по 6 ограничений. В случае полиномов меньшей степени система получится замкнутой (не получится подобрать полином, удовлетворяющий заданным ограничениям). В случае полиномов большей степени получится избыточная система (не получится явно задать коэффициенты полинома).

Допустим, что f(t) − полином пятой степени. Следовательно, для него должны выполняться ограничения:

$$f(0) = x_0, \dot{f}(0) = \dot{x}_0, \ddot{f}(0) = \ddot{x}_0, f(\Delta t) = x_*, \dot{f}(\Delta t) = \dot{x}_*, \ddot{f}(\Delta t) = \ddot{x}_*.$$

Запишем многочлен f(t) в виде:

$$x_i(t) = \sum_{j=0}^{2} \frac{x_{i0}^{(j)}}{j!} t^j + \sum_{j=1}^{3} c_{ij} t^{2+j}.$$

Очевидно, что при любых значениях параметра C_{ij}, полином удовлетворяет граничным условиям при t=0. Чтобы граничные условия выполнялись и при t=Δt, параметры C_{ij} должны удовлетворять системе, которую можно легко получить, записав соотношения условий, накладываемых на функцию в конечной точке траектории при t=Δt:

$$\begin{cases} (\Delta t)^3 c_{i0} + (\Delta t)^4 c_{i1} + (\Delta t)^5 c_{i2} = x_{i*} - x_{i0} - \dot{x}_{i0}(\Delta t) - 0.5\ddot{x}_{i0}(\Delta t)^2, \\ 3(\Delta t)^3 c_{i0} + 4(\Delta t)^4 c_{i1} + 5(\Delta t)^5 c_{i2} = \dot{x}_{i*} - \dot{x}_{i0} - \ddot{x}_{i0}(\Delta t), \\ 6(\Delta t)^3 c_{i0} + 12(\Delta t)^4 c_{i1} + 20(\Delta t)^5 c_{i2} = \ddot{x}_{i*} - \ddot{x}_{i0}. \end{cases} \qquad i = \overline{1,3}$$

Учитывая тот факт, что Δt≠0, можно сказать, что определитель матрицы СЛАУ равен $2(\Delta t)^9$, а следовательно решение этой системы линейных алгебраических уравнений существует, причём единственное. Найдя решение, а следовательно $x_1(t)$, $x_2(t)$, $x_3(t)$, можно будет определить функции L(t), Z(t) и H(t), которые определяют пространственную траекторию движения БПЛА, удовлетворяющую начальному и конечному условию задачи терминального управления.

ГЛАВА 3. ПОИСК ОПТИМАЛЬНОЙ ТРАЕКТОРИИ

В предыдущей главе было показано, как с помощью полиномов пятой степени можно получить пространственную траекторию БПЛА, удовлетворяющую наложенным на неё начальным и конечным условиям на промежутке времени t. Было показано, что время t − это единственный параметр, который характеризует вид полиномов, описывающих траекторию полёта в данной задаче.

Из естественных соображений следуют ограничения, накладываемые на переменные состояния и параметры управления летательного аппарата. Очевидно, что любой БПЛА имеет максимальную и минимальную скорость, ограничения высоты и т.д. Поэтому введём дополнительные ограничения на переменные состояния:

$$H_{min} \le H \le H_{max}; \quad L_{min} \le L \le L_{max}; \quad Z_{min} \le Z \le Z_{max};$$

$$V_{min} \le V \le V_{max}; \quad \vartheta_{min} \le \vartheta \le \vartheta_{max}; \quad \psi_{min} \le \psi \le \psi_{max};$$

и управления: $n_{Xmin} \le n_X \le n_{Xmax}$; $n_{Ymin} \le n_Y \le n_{Ymax}$; $\gamma_{min} \le \gamma \le \gamma_{max}$.

Поиск минимальной по времени движения БПЛА траектории с заданной точностью $\varepsilon > 0$ заключается в переборе по времени t (в порядке возрастания) траекторий, увеличивая время манёвра t на величину $\delta > 0$ (причём $\delta \ge \varepsilon$). Начальным значением для t (назовём его t_0) будем считать минимальное время, за которое можно перейти из начального состояния в конечное, двигаясь прямолинейно с максимально возможной скоростью:

$$t_0 = \frac{\sqrt{(H_* - H_0)^2 + (L_* - L_0)^2 + (Z_* - Z_0)^2}}{V_{max}}.$$

Конечным значением для перебора можно считать $t_{max} = (t_0 + 5)*15$.

Опытным путём было установлено, что полученного значения t_{max} было достаточно для максимального значения порога перебора, т.е., если искомое

время лежало в отрезке $[t_0; t_{max}]$, то оно находилось предложенным ниже алгоритмом, иначе условия, накладываемые на параметры и управления, не удовлетворялись.

Отметим, что на каждом шаге перебора необходимо проверять, удовлетворяет ли траектория и реализующие её управления условиям, которые накладываются на решение максимальными и минимальными значениями переменных состояния и управлений. Если была найдена удовлетворяющая граничным условиям траектория — то необходимо уменьшить соответствующее значение времени t на величину δ, а потом уменьшить значение δ в 2 раза (проводить уменьшение, пока δ не станет равным ε). Если была найдена траектория, не удовлетворяющая наложенным на неё ограничениям, то значение t необходимо увеличить на соответствующую величину δ.

Первая траектория, которая удовлетворяет этим ограничениям при δ=ε, и есть оптимальная по времени с точностью ε.

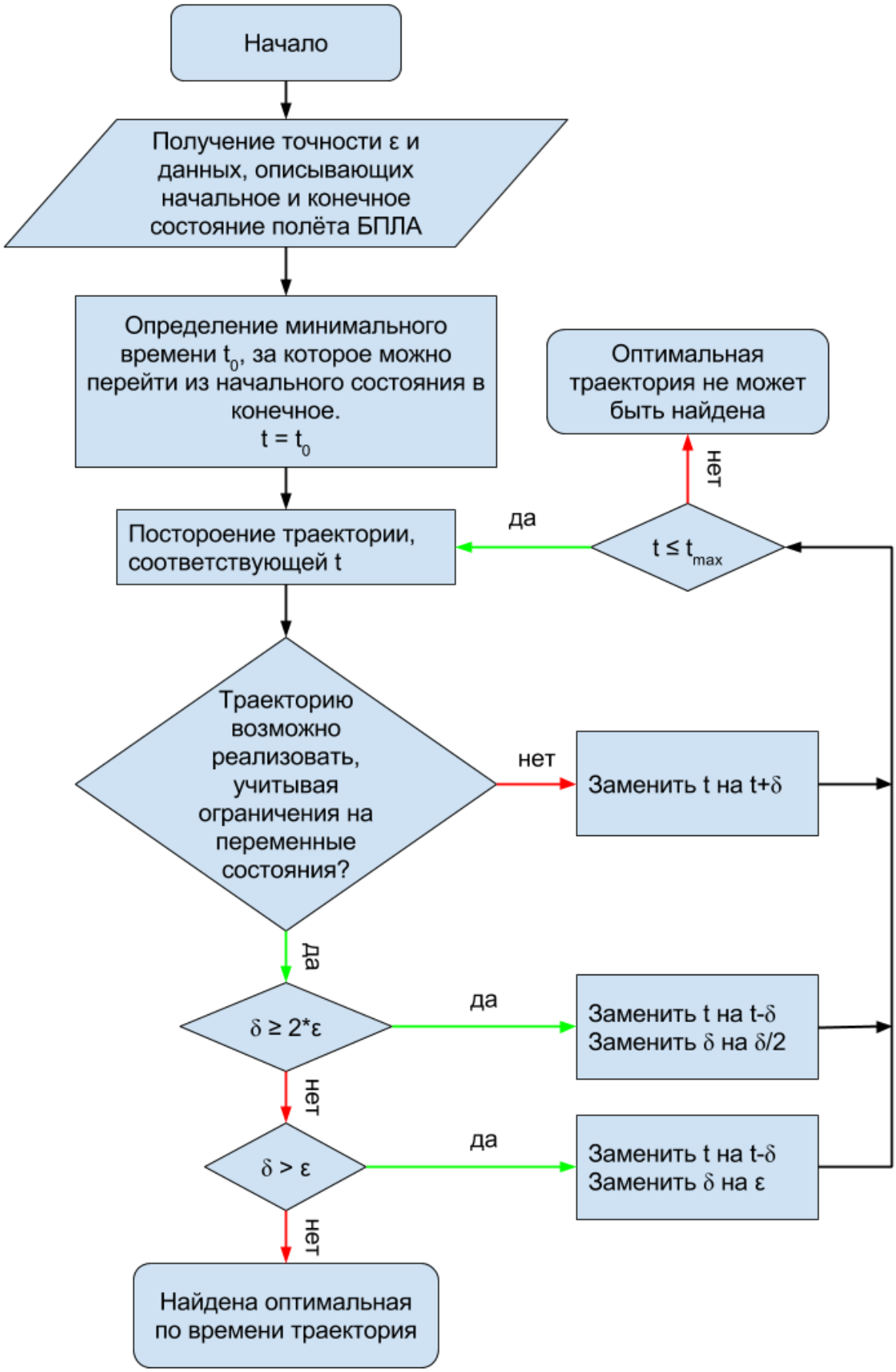

Рис. 4. Блок схема алгоритма поиска оптимальной траектории

ГЛАВА 4. РЕАЛИЗАЦИЯ ПРОГРАММНОГО КОМПЛЕКСА

Для реализации программного комплекса был выбран вычислительный пакет MATLAB в виду удобно реализованных математических операций и наличия огромного количества уже встроенных функций. Аргументом в пользу выбора MATLAB оказалось наличие инструментария разработки интерфейсов для приложений на базе MATLAB, а так же наличие встроенных утилит отрисовки графики.

4.1 Разработка интерфейса и функций в MATLAB

Графический интерфейс программного комплекса был реализован при помощи пакета GUIDE (Graphical User Interface Design Environment) [6].

Интерфейс состоит из:

- полей ввода ограничений на переменные состояния и управления БПЛА (заполнены заранее для удобства пользователя)
- полей ввода, характеризующих положение и управления БПЛА в начальной и конечной точке траектории
- кнопки, показывающей детальную информацию о переменных состояния
- кнопки, нажатие которой запустит алгоритм поиска оптимальной по времени траектории
- поля, сообщающего о неудаче поиска или о нахождении оптимальной траектории
- поля, показывающего полученное оптимальное время

- группы кнопок для отображения пространственной траектории полёта и графиков переменных, описывающих движение БПЛА по полученной траектории
- группы кнопок для отображения управлений, реализующих полученную траекторию.

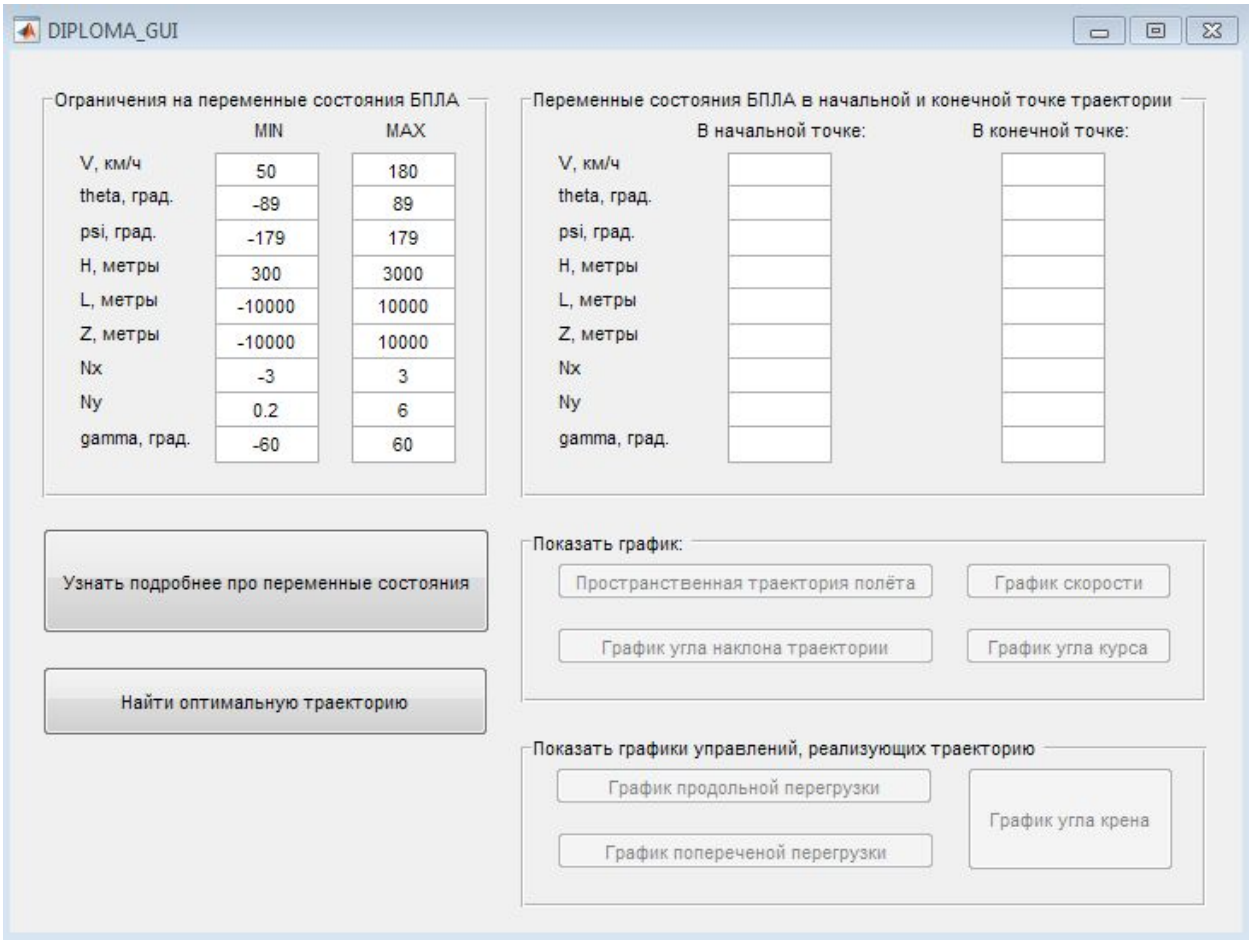

Рис. 5. Скриншот интерфейса приложения

При нажатии кнопки «Найти оптимальную траекторию» происходит следующее:

1. Считывание данных из полей ограничений и полей, описывающих начальное и конечное положение БПЛА.
2. Запуск реализованной .m функции «*optimum_finder*» (код функции содержится в приложении). Эта функция отвечает за создание переменных, необходимых для работы алгоритма

построения траектории, за преобразование переменных, создание логических флагов и за сохранение результатов работы. Если перебор по времени t завершился неудачей и траекторию невозможно было найти на отрезке времени [0; Δt], то функция возвращает логический флаг *flagTimeout = true*.

3. Далее «*optimum_finder*» в цикле по времени запускает .m функцию «*trajectory_finder*» (код функции содержится в приложении), которая в свою очередь уже занимается решением СЛАУ (процесс описан в главе 3), поиском численных значений полиномов, описывающих траекторию, поиском значений оставшихся переменных и управлений, необходимых для реализации траектории. Также эта функция проверяет полученную траекторию на то, удовлетворяет ли она заданным ограничениям и управляет алгоритмом поиска оптимального времени (описано в главе 4). Функция возвращает логический флаг *flagFound = true*, если процесс поиска оптимального значения t с точностью ε был завершён успешно.
4. После завершения выполнения алгоритма поиска оптимальной траектории, соответствующей полученному времени t, в случае нахождения траектории, сохраняются в рабочее пространство приложения все необходимые для графиков данные и выводится время, за которое должен быть совершён манёвр. В случае, когда траектория не может быть найдена, появляется соответствующая надпись-уведомление.

4.2 Тестирование работы программного комплекса в MATLAB

Для тестирования полученного программного комплекса в качестве ограничений для переменных состояния и управлений были взяты характеристики БПЛА российского производства Орлан-10 [7]. В частности были взяты минимальная и максимальная скорости, ограничения на высоту, перегрузки и угол крена. Ограничения на углы курса и наклона траектории были выбраны так, чтобы выполнялась гладкая замена переменных $\{|\vartheta|<\pi/2;\ |\psi|<\pi\}$. Ограничения на дальность полёта и боковое отклонение были выбраны так, чтобы БПЛА имел запас расстояния для выполнения сложных и длинных манёвров.

Список полученных ограничений выглядит так:

$300 \le H \le 5000$; $-10000 \le L \le 10000$; $-10000 \le Z \le 10000$

$75 \le V \le 170$; $-89 \le \vartheta \le 89$; $-179 \le \psi \le 179$

$-3 \le n_X \le 3$; $0.2 \le n_Y \le 6$; $-60 \le \gamma \le 60$

Далее приведён список манёвров, на которых был тестирован программный комплекс:

Манёвр 1. Разворот БПЛА на 90 градусов.

Значения переменных состояния и управлений в начальной и конечной точках траектории:

	H	L	Z	V	ϑ	ψ	n_x	n_y	γ
при t=0	900	0	0	120	0	0	0	1	0
при t=Δt	900	500	200	110	0	-90	0	1	0

Время, необходимое для того, чтобы совершить этот манёвр t = 15.988 с. Угол наклона траектории ϑ равен нулю на всём участке траектории.

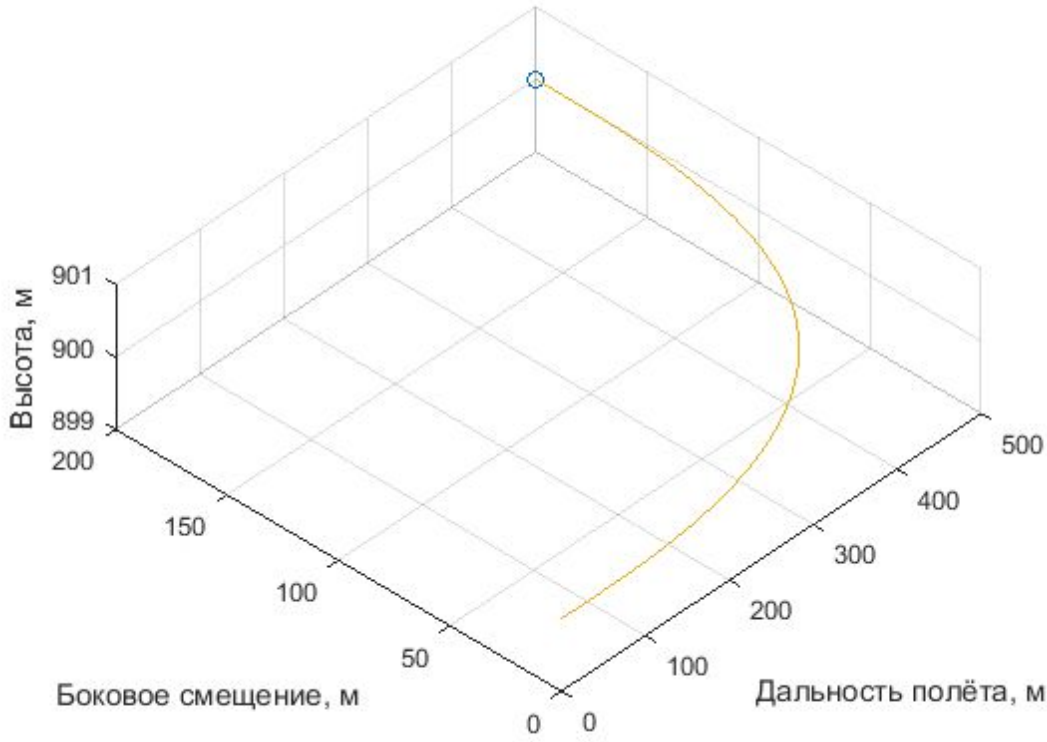

Рис. 6. Вид траектории в пространстве.

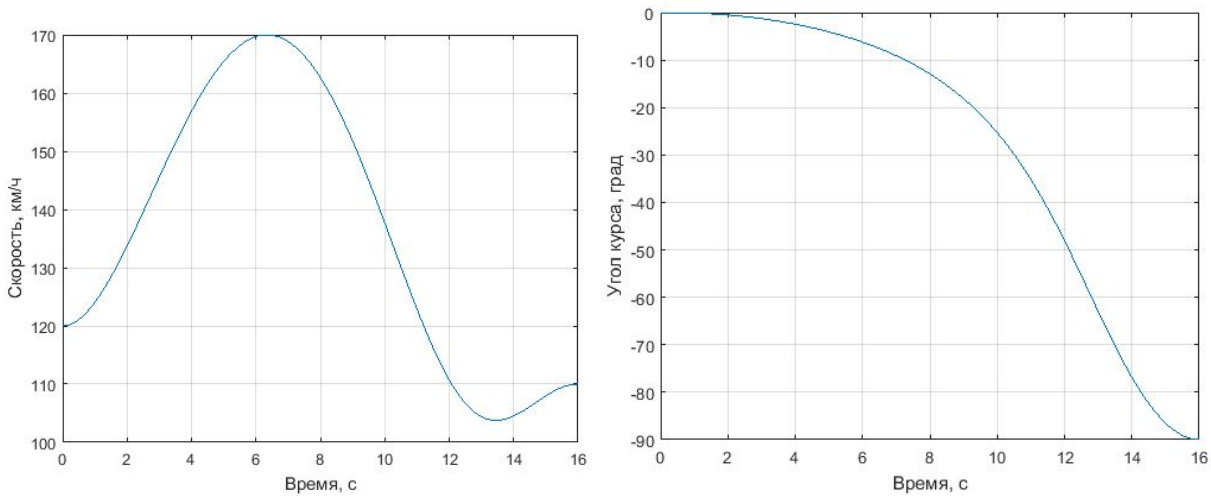

Рис. 7. График земной скорости БПЛА V и график курсового угла ψ.

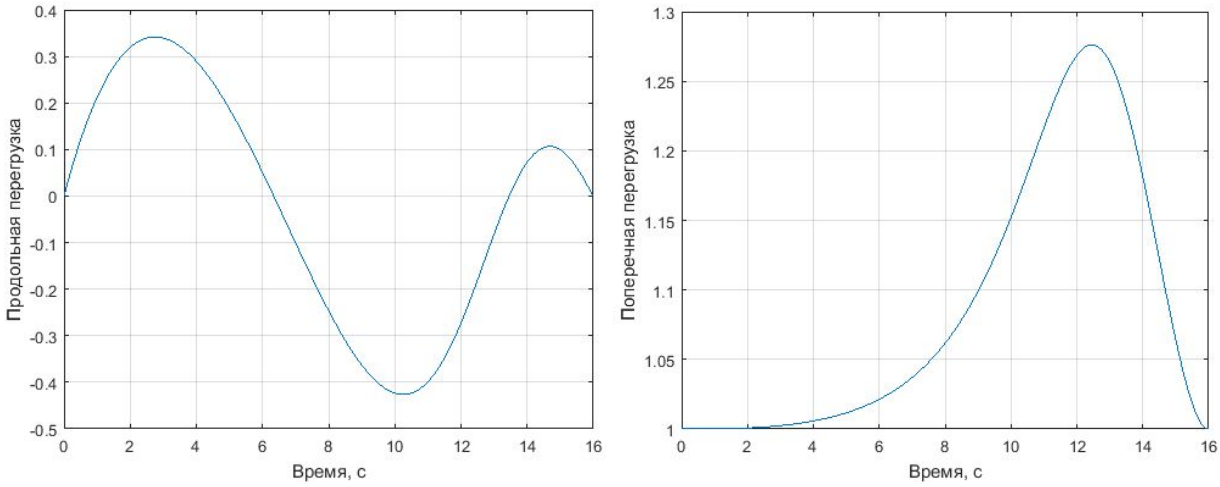

Рис. 8. Графики продольной перегрузки n_x и поперечной перегрузки n_y.

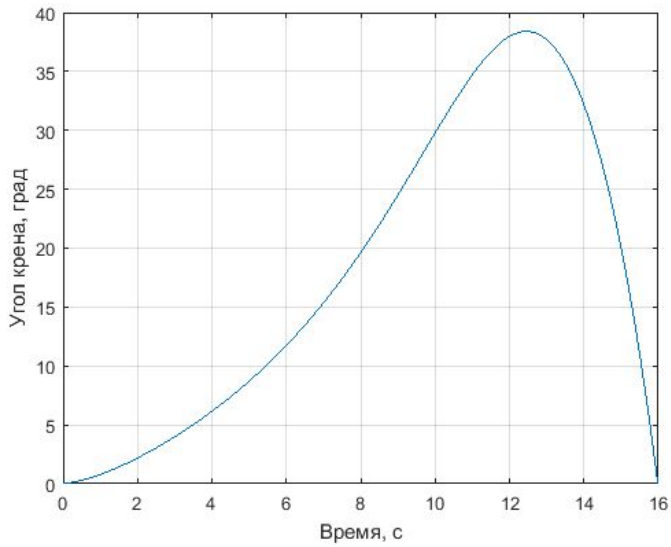

Рис. 9. График угла крена γ.

Манёвр 2. Смена эшелона (набор высоты).

Значения переменных состояния и управлений в начальной и конечной точках траектории:

	H	L	Z	V	ϑ	ψ	n_x	n_y	γ
при t=0	900	0	0	120	0	0	0	1	0
при t=Δt	1200	1000	0	110	0	0	0	1	0

Время, необходимое для того, чтобы совершить этот манёвр t = 26.7124 с. Курсовой угол ψ а также угол крена γ равны нулю на всём участке траектории.

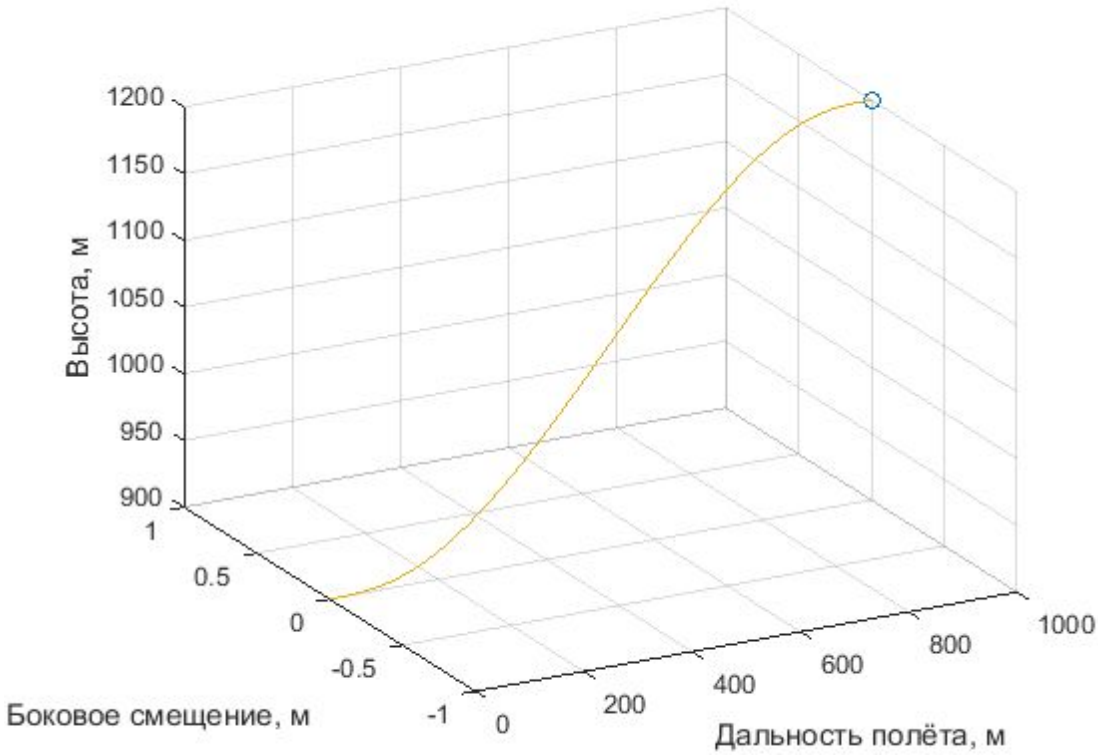

Рис. 10. Вид траектории в пространстве.

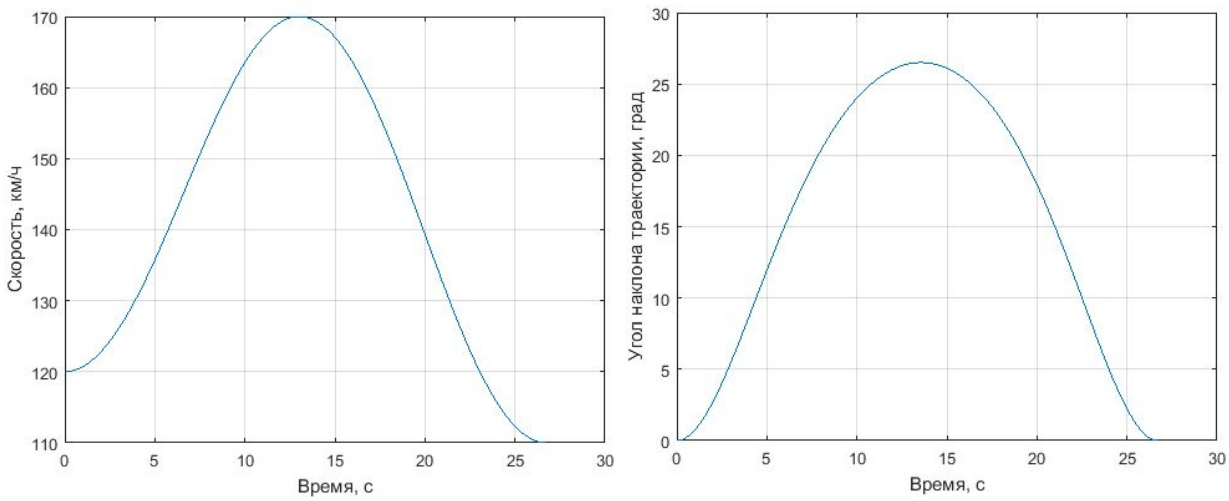

Рис. 11. График земной скорости БПЛА V и график угла наклона траектории ϑ.

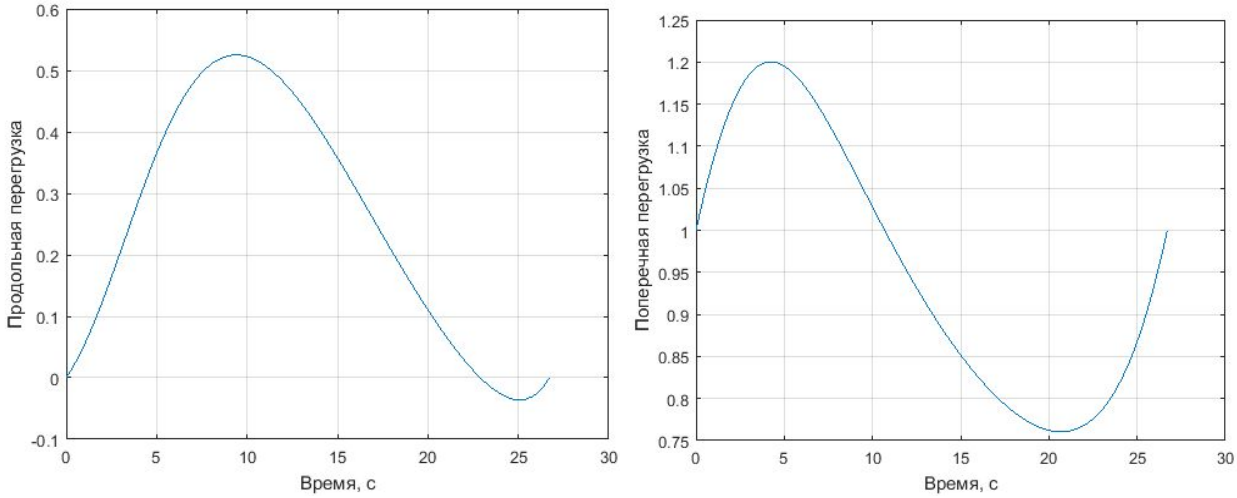

Рис. 12. Графики продольной перегрузки n_x и поперечной перегрузки n_y.

Манёвр 3. Уход от лобового столкновения (смена бокового смещения).

Значения переменных состояния и управлений в начальной и конечной точках траектории:

	H	L	Z	V	ϑ	ψ	n_x	n_y	γ
при t=0	1200	0	0	120	0	0	0	1	0
при t=Δt	1200	200	200	120	0	0	0	1	0

Время, необходимое для того, чтобы совершить этот манёвр t = 8.4741 с. Угол наклона траектории ϑ равен нулю на всём участке траектории.

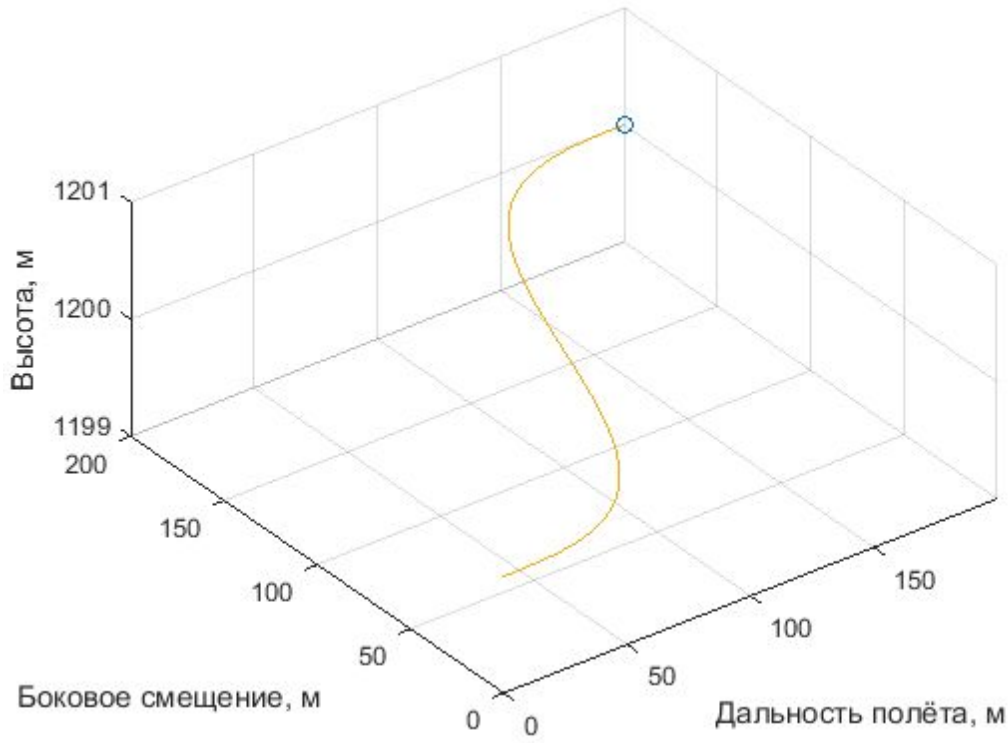

Рис. 13. Вид траектории в пространстве.

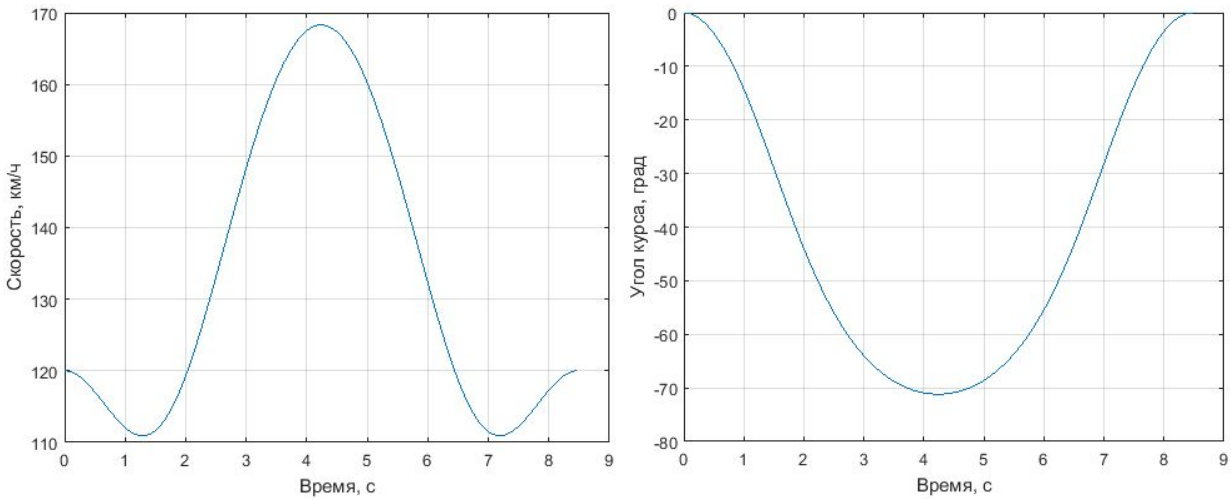

Рис. 14. График земной скорости БПЛА V и график курсового угла ψ.

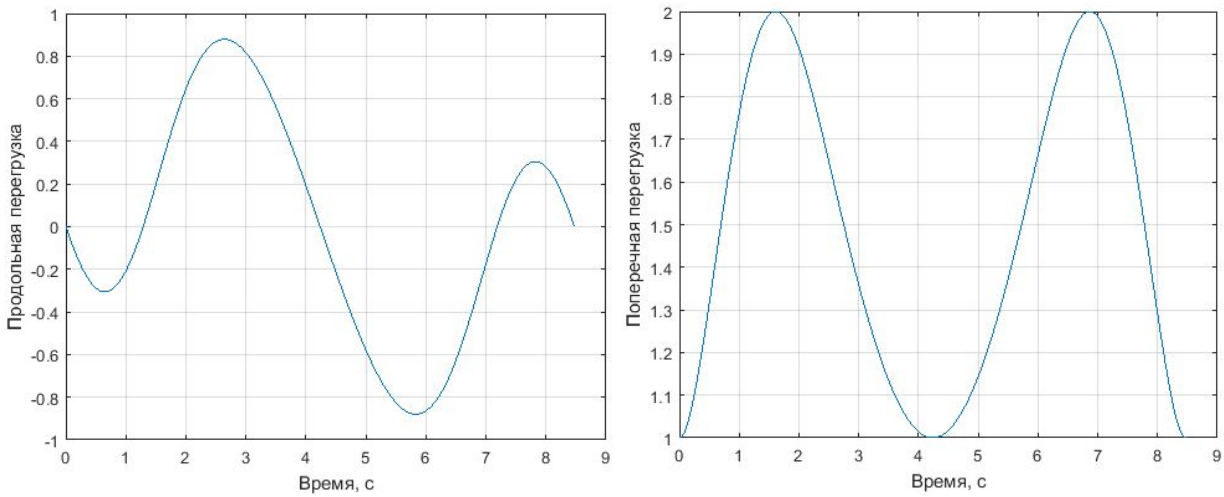

Рис. 15. Графики продольной перегрузки n_x и поперечной перегрузки n_y.

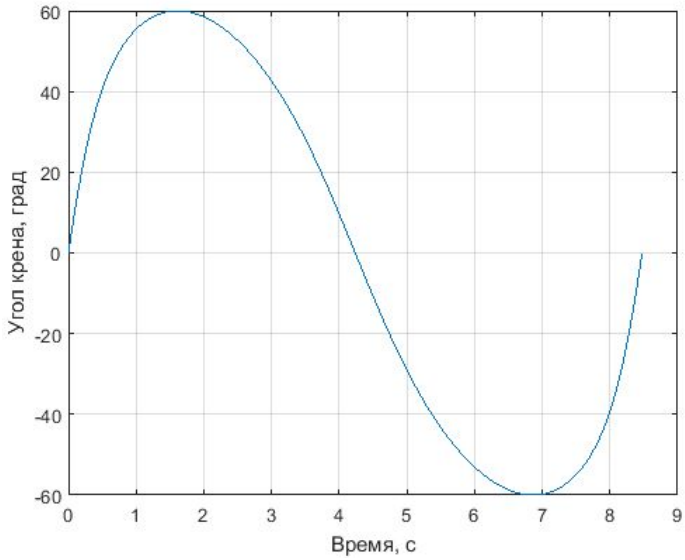

Рис. 16. График угла крена γ.

Манёвр 4. Разворот на 170 градусов и сброс высоты.

Значения переменных состояния и управлений в начальной и конечной точках траектории:

	H	L	Z	V	ϑ	ψ	n_x	n_y	γ
при t=0	1200	0	0	110	0	0	0	1	0
при t=Δt	900	0	300	130	0	-170	0	1	0

Время, необходимое для того, чтобы совершить этот манёвр t = 17.3959 с.

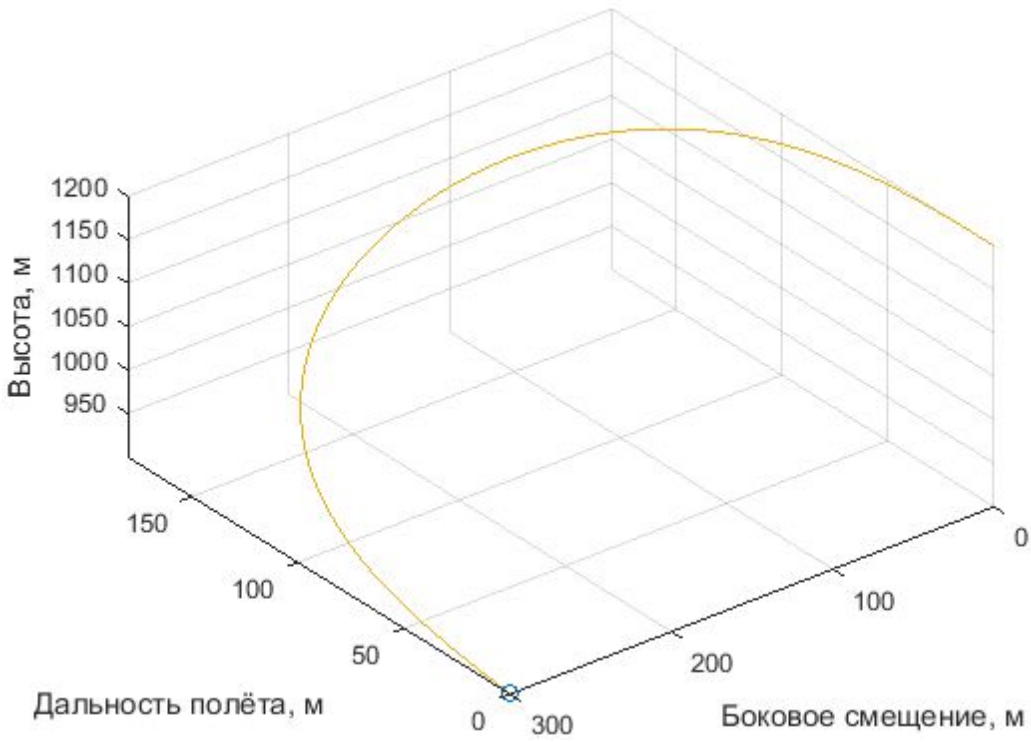

Рис. 17. Вид траектории в пространстве.

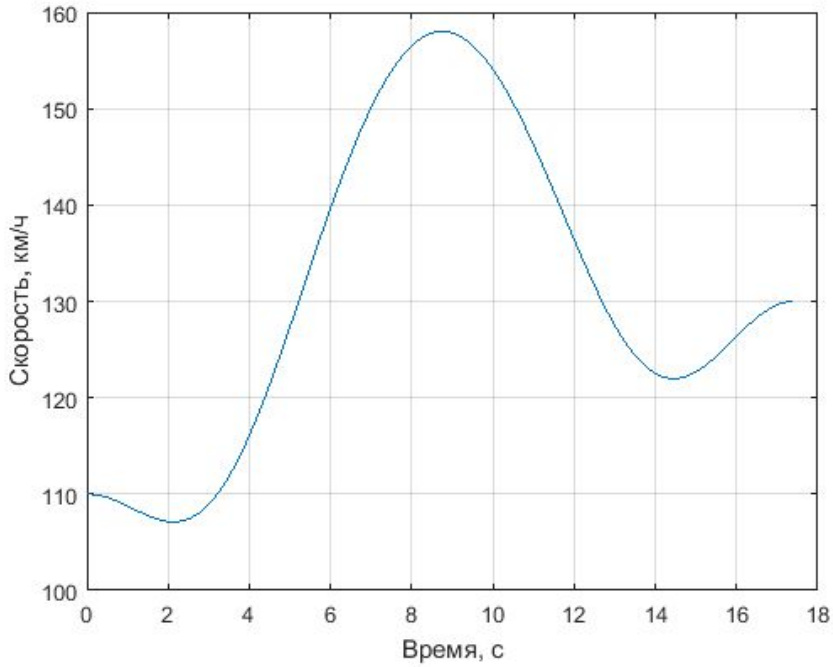

Рис. 18. График земной скорости БПЛА V

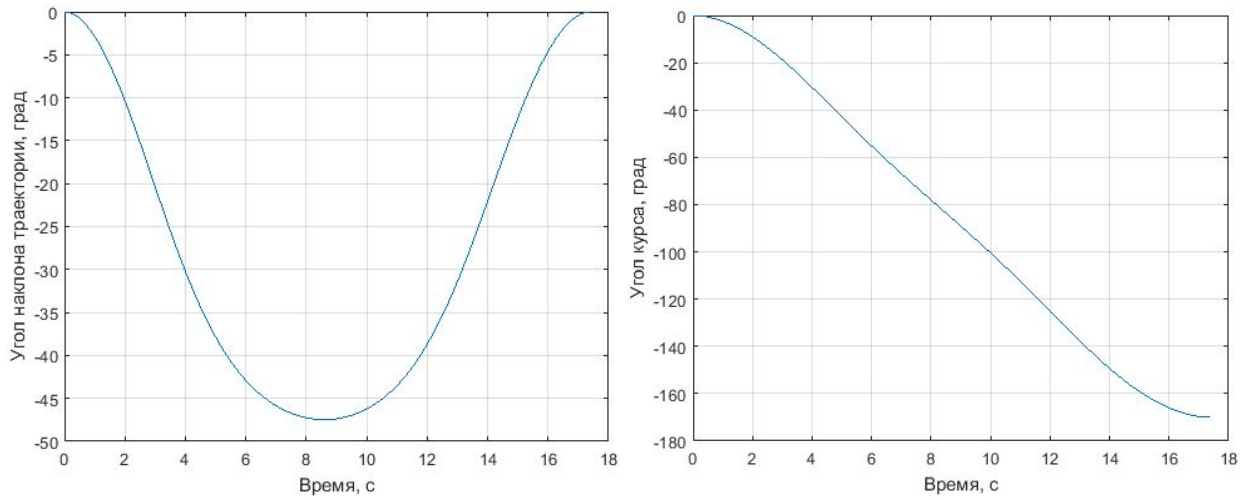

Рис. 19. График угла наклона траектории ϑ и график курсового угла ψ

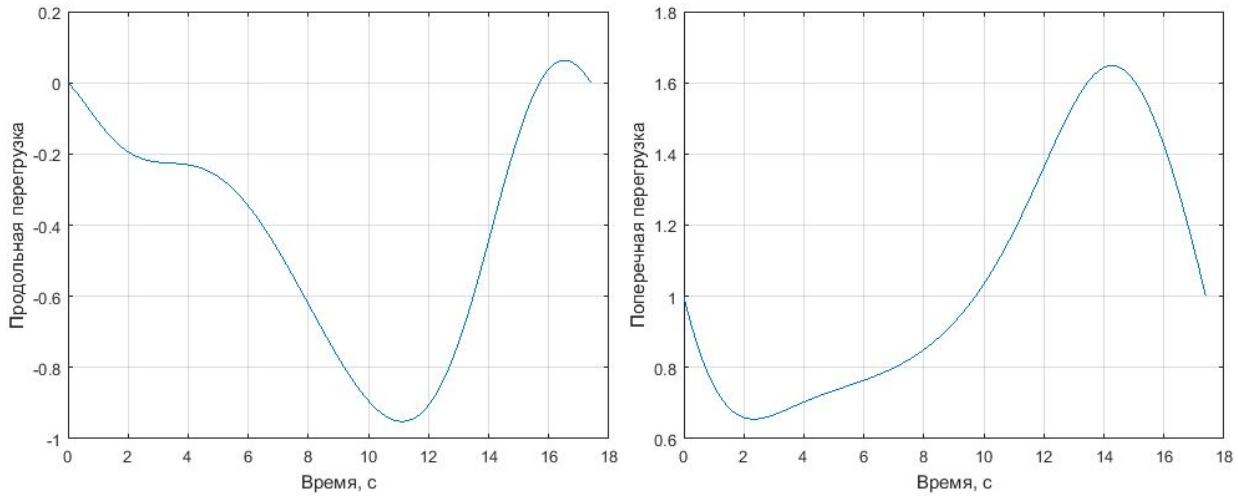

Рис. 20. Графики продольной перегрузки n_x и поперечной перегрузки n_y.

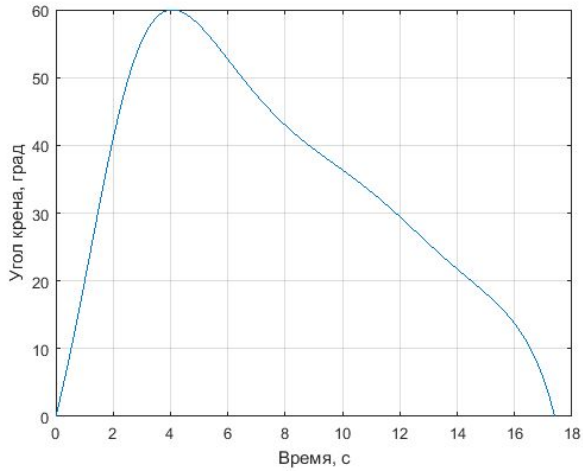

Рис. 21. График угла крена γ.

ВЫВОДЫ

В данной работе разработан программный комплекс построения оптимальной пространственной траектории БПЛА. В ходе выполнения ВКР была рассмотрена математическая модель управляемого движения БПЛА. Для построения траектории, описываемой начальными и конечными значениями переменных состояния, была рассмотрена задача терминального управления и представлены уравнения вычисления соответствующих необходимой траектории управлений. Для построения траектории движения БПЛА и решения поставленной задачи были использованы полиномы пятой степени. Был разработан и реализован программный комплекс в вычислительном пакете MATLAB r2016a, использующий алгоритм поиска оптимальной по времени траектории и реализующих её управлений.

Известно (см. введение), что алгоритмы, встроенные в бортовой компьютер БПЛА, должны работать в режиме реального времени. В среднем поиск с точностью до четвёртого знака после запятой выполнялся за 0.3 секунды. Зачастую, такая точность не является необходимой, а это значит, что можно достичь ещё большего быстродействия работы, снижая точность поиска (например, до двух знаков после запятой). Быстродействие с точностью до второго знака оказалось впечатляющим. В среднем поиск занимал 0.05 секунды, что может сказать о его пригодности в сфере управления БПЛА.

Тестирование полученного программного комплекса показало, что чаще всего БПЛА сталкивается с проблемой нехватки скорости для реализации ещё более быстрого манёвра. Реже реализация быстрейшей траектории упирается в ограничение летательного аппарата по углу крена. Наиболее редко встречалась проблема того, что значение перегрузок превышало допустимые пределы.

Работа программного комплекса показала, что он может использоваться для синтеза оптимальных траекторий движения БПЛА, причём в режиме реального времени.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы был создан программный комплекс, предлагающий оптимальную по времени траекторию полёта БПЛА при условии выполнения ограничений на переменные состояния и управляющие воздействия. Для этого был представлен обзор литературы по тематике исследования. Была описана математическая модель управляемого движения БПЛА, которую и было решено использовать впоследствии. Было представлено решение задачи терминального управления полётом БПЛА, разработан алгоритм поиска оптимальной по времени траектории. Создание программного комплекса в целом, а также моделирование алгоритма и тестирование полученного приложения было проведено в пакете MATLAB.

Полученные результаты могут быть использованы для создания алгоритмов возвращения БПЛА на базу в случае потери связи с оператором и для построения систем автоматического управления беспилотными летательными аппаратами. В дальнейшем данный программный комплекс может быть улучшен и дополнен путём внедрения функций управления с обратными связями, функций построения сложных траекторий, состоящих из базовых манёвров и др.

СПИСОК ЛИТЕРАТУРЫ

1. П.Д. Крутько. Обратная задача динамики управляемых систем: Нелинейные модели. – 1988
2. В.С. Моисеев. Теория управления беспилотными летательными аппаратами. – 2013.
3. А. Колесников. Новые нелинейные методы управления полетом. – 2013.
4. Т.С. Касаткина. Преобразования аффинных систем к каноническому виду с использованием замен независимой переменной. – 2013
5. Канатников А.Н., Шмагина Е.А. Задача терминального управления движением летательного аппарата // Нелинейная динамика и управление: Сборник статей. – 2010.
6. Онлайн документация вычислительного пакета MATLAB. http://www.mathworks.com/help/matlab/index.html
7. СТЦ Орлан-10 // Авиационная энциклопедия. http://www.airwar.ru/enc/bpla/orlan10.html

ПРИЛОЖЕНИЕ

Листинг кода:

Код функции ***optimum_finder.m:***

```
Time_0 = sqrt((H_Fin-H_0)^2+(L_Fin-L_0)^2+(Z_Fin-Z_0)^2)/(V_MAX/3.6);
Time_Fin = (Time_0+5)*15;
deltaTime = 0.5;
precision = 0.0001;
flagFound = false;
flagTimeout = false;
curTime = Time_0;
g = 9.80665;
trueVector = true(1, 1001);
% START VIRTUAL CONTROL
v_0 = [Nx_0
       Ny_0*cosd(gamma_0)
       Ny_0*sind(gamma_0)];
% FINAL VIRTUAL CONTROL
v_Fin = [Nx_Fin
         Ny_Fin*cosd(gamma_Fin)
         Ny_Fin*sind(gamma_Fin)];
% SPEED INTO METERS PER SECOND
V_0 = V_0/3.6;
V_Fin = V_Fin/3.6;
% START CONDITION IN NEW TERMS
y10 = H_0;
y20 = L_0;
y30 = Z_0;
y10f = V_0*sind(theta_0);
y20f = V_0*cosd(theta_0)*cosd(psi_0);
y30f = -1*V_0*cosd(theta_0)*sind(psi_0);
y10s = -g + g*v_0(1)*sind(theta_0) + g*v_0(2)*cosd(theta_0);
y20s = g*v_0(1)*cosd(theta_0)*cosd(psi_0) - g*v_0(2)*sind(theta_0)*cosd(psi_0) +
g*v_0(3)*sind(psi_0);
y30s = -g*v_0(1)*cosd(theta_0)*sind(psi_0) + g*v_0(2)*sind(theta_0)*sind(psi_0) +
g*v_0(3)*cosd(psi_0);
% FINAL CONDITION IN NEW TERMS
y1F = H_Fin;
y2F = L_Fin;
y3F = Z_Fin;
y1Ff = V_Fin*sind(theta_Fin);
y2Ff = V_Fin*cosd(theta_Fin)*cosd(psi_Fin);
```

```
y3Ff = -1*V_Fin*cosd(theta_Fin)*sind(psi_Fin);
y1Fs = -g + g*v_Fin(1)*sind(theta_Fin) + g*v_Fin(2)*cosd(theta_Fin);
y2Fs = g*v_Fin(1)*cosd(theta_Fin)*cosd(psi_Fin) -
g*v_Fin(2)*sind(theta_Fin)*cosd(psi_Fin) + g*v_Fin(3)*sind(psi_Fin);
y3Fs = -g*v_Fin(1)*cosd(theta_Fin)*sind(psi_Fin) +
g*v_Fin(2)*sind(theta_Fin)*sind(psi_Fin) + g*v_Fin(3)*cosd(psi_Fin);

while ((not(flagFound))&&(curTime<=Time_Fin))
    trajectory_finder;
end

if(curTime>Time_Fin)
    flagTimeout = true;
end
```

Код функции ***trajectory_finder.m:***

```
time = 0:curTime/1000:curTime;
delta = curTime;
% FINDING COEFFICIENTS OF POLINOMIAL
A = [delta^3 delta^4 delta^5;
     3*delta^2 4*delta^3 5*delta^4;
     6*delta 12*delta^2 20*delta^3];
B1 = [y1F - y10 - delta*y10f - 0.5*delta^2*y10s;
      y1Ff - y10f - delta*y10s;
      y1Fs - y10s];
B2 = [y2F - y20 - delta*y20f - 0.5*delta^2*y20s;
      y2Ff - y20f - delta*y20s;
      y2Fs - y20s];
B3 = [y3F - y30 - delta*y30f - 0.5*delta^2*y30s;
      y3Ff - y30f - delta*y30s;
      y3Fs - y30s];
C1 = linsolve(A, B1)';
C2 = linsolve(A, B2)';
C3 = linsolve(A, B3)';
% BUILDING A TRAJECTORY
H = y10 + y10f*time + 0.5*y10s*time.^2 + C1(1)*time.^3 + C1(2)*time.^4 + C1(3)*time.^5;
L = y20 + y20f*time + 0.5*y20s*time.^2 + C2(1)*time.^3 + C2(2)*time.^4 + C2(3)*time.^5;
Z = y30 + y30f*time + 0.5*y30s*time.^2 + C3(1)*time.^3 + C3(2)*time.^4 + C3(3)*time.^5;
% CALCULATING OTHER POLYNOMIALS
V = sqrt((y10f + y10s*time + 3*C1(1)*time.^2 + 4*C1(2)*time.^3 + 5*C1(3)*time.^4).^2 +
(y20f + y20s*time + 3*C2(1)*time.^2 + 4*C2(2)*time.^3+ 5*C2(3)*time.^4).^2 + (y30f +
y30s*time + 3*C3(1)*time.^2 + 4*C3(2)*time.^3 + 5*C3(3)*time.^4).^2);
sinTheta = (y10f + y10s*time + 3*C1(1)*time.^2 + 4*C1(2)*time.^3 + 5*C1(3)*time.^4)./V;
```

```
theta = asind(sinTheta);
cosTheta = cosd(theta);
sinPsi = -(y30f + y30s*time + 3*C3(1)*time.^2 + 4*C3(2)*time.^3 + 5*C3(3)*time.^4)./sqrt((y20f + y20s*time + 3*C2(1)*time.^2 +4*C2(2)*time.^3 + 5*C2(3)*time.^4).^2 + (y30f + y30s*time + 3*C3(1)*time.^2 + 4*C3(2)*time.^3 + 5*C3(3)*time.^4).^2);
cosPsi = (y20f + y20s*time + 3*C2(1)*time.^2 + 4*C2(2)*time.^3 + 5*C2(3)*time.^4)./sqrt((y20f + y20s*time + 3*C2(1)*time.^2 +4*C2(2)*time.^3 + 5*C2(3)*time.^4).^2 + (y30f + y30s*time + 3*C3(1)*time.^2 + 4*C3(2)*time.^3 + 5*C3(3)*time.^4).^2);
psy = atan2d(sinPsi, cosPsi);
v1 = ((y10s + 6*C1(1)*time + 12*C1(2)*time.^2 + 20*C1(3)*time.^3 + g).*sinTheta + (y20s + 6*C2(1)*time + 12*C2(2)*time.^2 +20*C2(3)*time.^3).*cosTheta.*cosPsi - (y30s + 6*C3(1)*time + 12*C3(2)*time.^2 + 20*C3(3)*time.^3).*cosTheta.*sinPsi) / g;
v2 = ((y10s + 6*C1(1)*time + 12*C1(2)*time.^2 + 20*C1(3)*time.^3 + g).*cosTheta - (y20s + 6*C2(1)*time + 12*C2(2)*time.^2 +20*C2(3)*time.^3).*sinTheta.*cosPsi + (y30s + 6*C3(1)*time + 12*C3(2)*time.^2 + 20*C3(3)*time.^3).*sinTheta.*sinPsi) / g;
v3 = ((y20s + 6*C2(1)*time + 12*C2(2)*time.^2 + 20*C2(3)*time.^3).*sinPsi + (y30s + 6*C3(1)*time + 12*C3(2)*time.^2 +20*C3(3)*time.^3).*cosPsi) / g;
tanGamma = v3./v2;
gama = atand(tanGamma);
cosGamma = cosd(gama);
Nx = v1;
Ny = v2./cosGamma;
% CONVERTING SPEED BACK INTO KM/H
V = V*3.6;
if isequal((V >= V_MIN)&(V <= V_MAX)&(theta >= theta_MIN)&(theta <= theta_MAX)&(psy >= psi_MIN)&(psy <= psi_MAX)&(H >= H_MIN)&(H <= H_MAX)&(L >= L_MIN)&(L <= L_MAX)&(Z >= Z_MIN)&(Z <= Z_MAX)&(Nx >= Nx_MIN)&(Nx <= Nx_MAX)&(Ny >= Ny_MIN)&(Ny <= Ny_MAX)&(gama >= gamma_MIN)&(gama <= gamma_MAX), trueVector)
    if (deltaTime >= 2*precision)
        curTime = curTime - deltaTime;
        deltaTime = deltaTime/2;
    elseif (deltaTime > precision)
        curTime = curTime - deltaTime;
        deltaTime = precision;
    else
        flagFound = true;
    end
else
    curTime = curTime + deltaTime;
end
```