

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
КАФЕДРА МЕХАНИКИ УПРАВЛЯЕМОГО ДВИЖЕНИЯ

Тимченко Богдан Павлович

Магистерская диссертация

# Исследование задачи посадки БПЛА на судно-носитель

Направление 01.04.02

Прикладная математика и информатика

Математическое моделирование в задачах естествознания

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

Санкт-Петербург  
2018

# Содержание

Введение . . . . .	3
Обзор литературы . . . . .	5
Глава 1. Математическая модель движения летательного аппарата . . . . .	6
1.1. Определение положения летательного аппарата . . . . .	6
1.2. Определение направления движения летательного аппарата . . . . .	6
1.3. Описание модели движения летательного аппарата . . . . .	8
Постановка задачи . . . . .	12
Глава 2. Задача терминального управления полетом летательного аппарата . . . . .	13
2.1. Решение задачи терминального управления . . . . .	13
Глава 3. Построение условно-оптимальной траектории посадки летательного на движущееся судно . . . . .	16
3.1. Построение пространственной траектории летательного аппарата . . . . .	16
3.2. Поиск условно-оптимальной траектории летательного аппарата . . . . .	17
3.3. Определение положения судна в момент посадки . . . . .	18
3.4. Поиск условно-оптимальной траектории посадки летательного на движущееся судно . . . . .	19
Результаты тестирования программного комплекса . . . . .	21
Выводы . . . . .	31
Заключение . . . . .	32
Список литературы . . . . .	33
Приложение . . . . .	34

# Введение

Беспилотные летательные аппараты играют все более значимую роль в военных целях. Они решают задачи разведки, наблюдения и коммуникации, при длительных полетах и в экстремальных условиях. Однако также БПЛА успешно применяются и для решения многих гражданских задач, предоставляя широкие возможности, такие как контроль пожарной безопасности и предупреждение стихийных бедствий, патрулирование приграничных зон, наблюдение за территориями заповедников и другие.

Использование БПЛА для наблюдения за состоянием водной поверхности и прилегающих к ней территорий является наиболее перспективным направлением с точки зрения целесообразности применения судовых комплексов военного и гражданского назначения.

Сравнение БПЛА различных типов показало ряд преимуществ БПЛА самолетного типа: большее время полета, больший радиус действия, больший коэффициент полезной нагрузки, лучшие аэродинамические показатели. Однако БПЛА вертолетного типа имеют серьезное преимущество – они могут совершать посадку на площадки ограниченных размеров, на что не способны БПЛА самолетного типа. Размещение и применение многоцелевых комплексов БПЛА на малых судах реализуется только тогда, когда гарантировано обеспечение их безопасной посадки.

Вследствие эксплуатационных и технических особенностей, с которыми сталкивается флот при использовании БПЛА на судах гражданского назначения, не приспособленных для посадки летательных аппаратов, особое значение имеет обеспечение возможности размещения на судне требуемых средств посадки. Например, обеспечение взлетно-посадочной полосы, на которую осуществляется посадка БПЛА с последующим пробегом и торможением. Но решение этого вопроса часто связано с необходимостью кардинального изменения архитектуры судна для поиска необходимых площадей. Это не всегда является возможным и приемлемым в связи с вероятными существенными изменениями некоторых эксплуатационных и технических характеристик судна.

Наиболее трудоемкой задачей является посадка «сухим» методом, поскольку после посадки на воду, необходим восстановительный ремонт ле-

тательного аппарата в результате воздействия воды, что связано со значительными затратами. Вследствие этого отрабатываются различные методы посадки БПЛА на палубу судна, такие как: посадка БПЛА в вертикальную сеть, подхват стабилизируемой кран-балкой размещенной на борту судна и др.

Обычно в БПЛА внедряют специальное программное обеспечение, реализующее автоматическое управление аппаратом. Система автопилота полностью контролирует выбор траектории, управление тягой двигателя и положением рулей. Данный способ управления предпочтителен, так как исключает воздействие человеческого фактора, а также позволяет избежать аварий, происходящих из-за потери связи с центром управления.

Все вышесказанное приводит к необходимости создания алгоритма, реализующего в автоматическом режиме построение траектории, производящей посадку БПЛА из некоторого начального положения на палубу движущегося судна. Важно, чтобы построенная траектория была оптимальной, так как сокращение расхода топлива и времени полета являются наиболее важными условиями эффективного использования БПЛА.

## Обзор литературы

В связи с быстрыми темпами развития в области управления беспилотными летательными аппаратами, за прошедшие годы было написано большое количество научных трудов, описывающих основные понятия исследований данного направления.

В книге П.Д. Крутько «Обратная задача динамики управляемых систем: Нелинейные модели» [1] приведено подробное описание движения летательного аппарата в различных плоскостях, а также его математическая модель описываемая дифференциальными уравнениями. Материал из этой книги был использован в качестве основы для разработки математической модели в данной работе. Однако полная математическая модель оказалась крайне сложной для вычисления. Для упрощения был использован материал из работы В.С. Моисеева «Теория управления беспилотными летательными аппаратами» [2], а также книга А. Колесникова «Новые нелинейные методы управления полетом» [3]. В результате была получена упрощённая математическая модель, которая была использована в дальнейшем.

Информация о преобразованиях линейных систем к каноническому виду рассмотрена в работе Т.С. Касаткиной «Преобразования аффинных систем к каноническому виду с использованием замен независимой переменной» [4].

В статье А.Н. Канатникова и Е.А. Шмагиной «Задача терминального управления движением летательного аппарата» [5], описан метод построения пространственной траектории движения летательного аппарата и способ расчёта соответствующих управлений.

Для реализации программного комплекса и поиска оптимальной траектории была использована онлайн документация к математическому пакету Wolfram Mathematica 11.2 [6].

# Глава 1. Математическая модель движения летательного аппарата

## 1.1. Определение положения летательного аппарата

Стандартным методом для определения координат БПЛА является описание положения его центра масс относительно Земли, определяемое в нормальной земной системе координат. Под нормальной земной системой координат будем понимать правую декартову прямоугольную систему координат  $OXYZ$ , которая является инерциальной и жестко связана с Землей [3]. Началом этой системы координат в некоторой неподвижной точке  $O$ , которая расположена на земной поверхности. Ось  $OY$  направлена вертикально в противоположную сторону от вектора силы тяжести. Земля в данной системе представлена, как плоскость, образуемая осями координат  $X$  и  $Z$ , при  $Y = 0$ .

Таким образом, положение центра масс БПЛА относительно Земли можно описать, задавая координаты  $(X, Y, Z)$ . Координата  $X$  характеризует дальность полёта  $L$ , координата  $Y$  описывает высоту полёта  $H$ , а координата  $Z$  характеризует боковое смещение  $Z$ .

## 1.2. Определение направления движения летательного аппарата

Направлением движения БПЛА будем принимать ориентацию его вектора скорости  $V$ . Чтобы определить направление движения рассмотрим траекторную систему координат  $O_tX_tY_tZ_t$  [5]. Эта система координат жестко связана с центром масс БПЛА, причем её начало  $O_t$  расположено в центре масс летательного аппарата. Ось  $O_tX_t$  направлена вдоль земной скорости БПЛА, ось  $O_tY_t$  перпендикулярна оси  $O_tX_t$  и направлена вверх относительно местной вертикали, а ось  $O_tZ_t$  дополняет их до правой тройки векторов.

Таким образом, определение направления движения БПЛА происходит путем ориентации осей траекторной системы координат относительно нормальной земной системы координат. Положение траекторной системы

координат задаётся углом наклона траектории  $\vartheta$  и курсовым углом  $\psi$ .

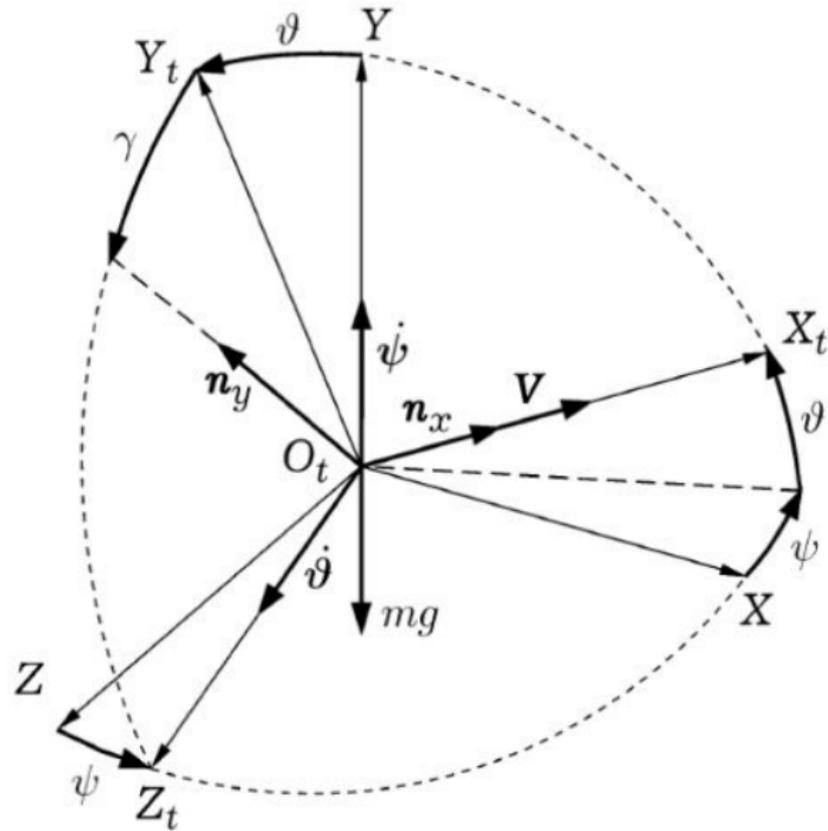


Рис. 1: Нормальная земная и траекторная система координат. На рисунке показан вектор скорости  $V$ , угол наклона траектории  $\vartheta$ , курсовой угол  $\psi$ , угол крена  $\gamma$  и перегрузки  $n_x$  и  $n_y$

### Условные обозначения физической модели БПЛА

$H$  – высота полёта

$L$  – дальность полёта

$Z$  – боковое смещение

$V$  – путевая скорость

$\vartheta$  – угол наклона траектории

$\psi$  – путевой угол

$\gamma$  – угол крена

$n_x$  – продольная перегрузка

$n_Y$  – поперечная перегрузка

$g$  - ускорение свободного падения

### 1.3. Описание модели движения летательного аппарата

Летательный аппарат рассматривается как свободное твёрдое тело, имеющее шесть степеней свободы. Рассмотрим динамические уравнения движения центра масс летательного аппарата в траекторной системе координат [1].

$$\begin{cases} m\dot{V} = \sum F_{X_t}, \\ mV\dot{\vartheta} = \sum F_{Y_t}, \\ -mV\dot{\psi}\cos(\vartheta) = \sum F_{Z_t}. \end{cases} \quad (1)$$

Силы, действующие на летательный аппарат в процессе полёта:

$mg$  – сила тяжести

$P$  – сила тяги

$X'$  – сила лобового сопротивления

$Y'$  – подъёмная сила

$Z'$  – боковая сила

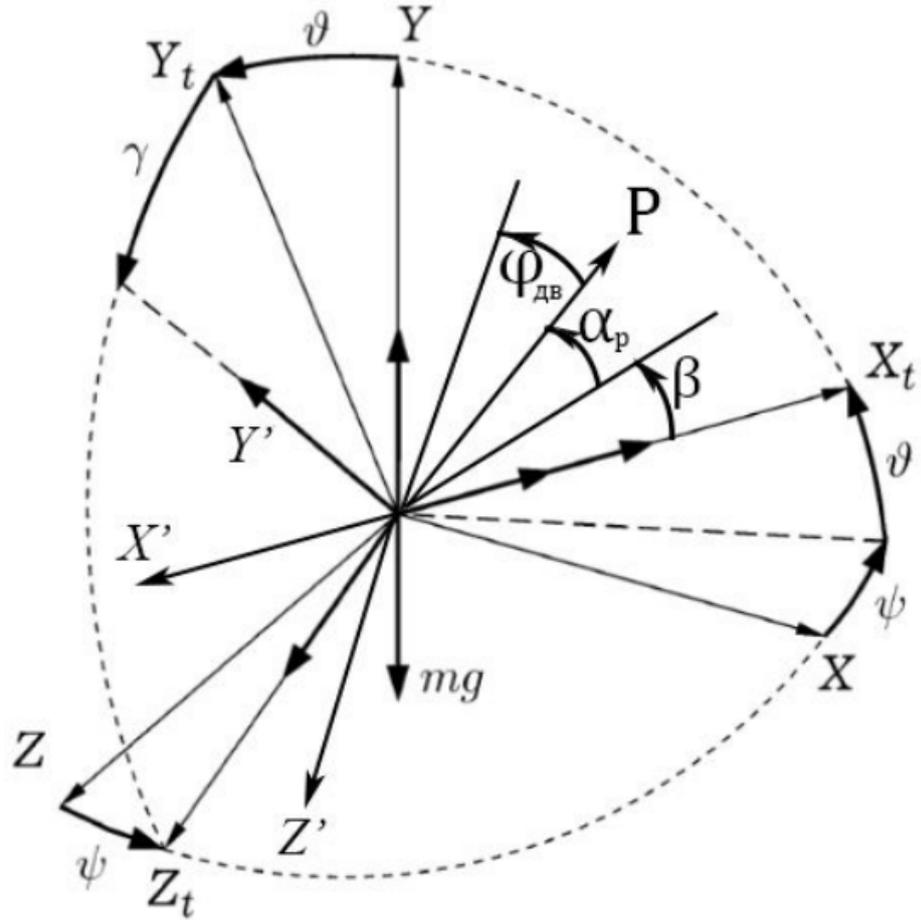


Рис. 2: Нормальная земная и траекторная система координат с отмеченными векторами сил, приложенных к БПЛА

Из Рис.2, можно получить проекции всех указанных выше сил на оси траекторной системы координат. В итоге для уравнений (1) получим следующие равенства:

$$\begin{cases} m\dot{V} = PCos(\alpha_p)Cos(\beta) - X' - mgSin(\vartheta), \\ mV\dot{\vartheta} = (Y' + PSin(\alpha_p))Cos(\gamma) - (Z' - PCos(\alpha_p)Sin(\beta))Sin(\gamma) - mgCos(\vartheta), \\ -mV\dot{\psi}Cos(\vartheta) = (Y' + PSin(\alpha_p))Sin(\gamma) + (Z' - PCos(\alpha_p)Sin(\beta))Cos(\gamma). \end{cases} \quad (2)$$

При решении таких расчетных практических задач, как построение траектории, будут справедливы следующие приближенные равенства:

$$Sin\alpha \approx \alpha, Sin\beta \approx \beta, Cos\alpha \approx 1, Cos\beta \approx 1.$$

Тогда, система уравнений (2) может быть записана следующим обра-

ЗОМ:

$$\begin{cases} \dot{V} = \frac{P-X'}{m} - g\text{Sin}(\vartheta), \\ \dot{\vartheta} = \frac{(Y'+P\alpha_p)\text{Cos}(\gamma)-(Z'-P\beta)\text{Sin}(\gamma)}{mV} - \frac{g}{V}\text{Cos}(\vartheta), \\ \dot{\psi} = -\frac{(Y'+P\alpha_p)\text{Sin}(\gamma)+(Z'-P\beta)\text{Sin}(\gamma)}{mV\text{Cos}(\vartheta)}. \end{cases} \quad (3)$$

В этой работе рассматривается способ бокового маневрирования, при котором нужная боковая сила возникает за счёт изменения угла крена. В таком случае при построении модели принимают  $\beta = 0, Z' = 0$  [2]. Таким образом, подставляя данные значения в систему (3), получим следующую систему дифференциальных уравнений:

$$\begin{cases} \dot{V} = \frac{P-X'}{m} - g\text{Sin}(\vartheta), \\ \dot{\vartheta} = \frac{(Y'+P\alpha_p)\text{Cos}(\gamma)}{mV} - \frac{g}{V}\text{Cos}(\vartheta), \\ \dot{\psi} = -\frac{(Y'+P\alpha_p)\text{Sin}(\gamma)}{mV\text{Cos}(\vartheta)}. \end{cases} \quad (4)$$

При исследовании движения БПЛА часто используют понятие перегрузок, через которые выражают динамические уравнения движения. Компоненты вектора перегрузки выражаются следующим образом:

$$n_X = \frac{P - X'}{mg}, n_Y = \frac{Y' + P\alpha_p}{mg}$$

Подставляя эти выражения в (4), получим следующую систему динамических уравнений в перегрузках:

$$\begin{cases} \dot{V} = g(n_X - \text{Sin}(\vartheta)), \\ \dot{\vartheta} = g\frac{n_Y\text{Cos}(\gamma)-\text{Cos}(\vartheta)}{V}, \\ \dot{\psi} = -g\frac{n_Y\text{Sin}(\gamma)}{V\text{Cos}(\vartheta)}. \end{cases} \quad (5)$$

Для упрощения модели динамические уравнения вращения летательного аппарата в данной работе рассмотрены не будут. Далее будем считать, что при рассматриваемом управлении БПЛА время переходных процессов по каналам крена, тангажа и рысканья должно быть малым. Исходя из этого можно сделать предположение о том, что углы крена  $\gamma$ , атаки  $\alpha$  и скольжения  $\beta$  устанавливаются мгновенно при изменении положений соответствующих элементов управления БПЛА. Такая идеализированная

система управления летательным аппаратом позволяет считать его безынерционным вращающимся объектом, для которого выполнены условия равенства нулю главных центральных моментов инерции, и, следовательно, оказываются равны нулю суммы проекций моментов действующих сил на БПЛА относительно его центра масс.

Положение центра масс летательного аппарата относительно нормальной земной системы координат определяется следующими уравнениями [1]:

$$\begin{cases} \dot{H} = V \sin(\vartheta), \\ \dot{L} = V \cos(\vartheta) \cos(\psi), \\ \dot{Z} = -V \cos(\vartheta) \sin(\psi). \end{cases} \quad (6)$$

Объединив системы (5) и (6), мы получим математическую модель, в виде системы из шести дифференциальных уравнений, описывающих положение и движение центра масс летательного аппарата:

$$\begin{cases} \dot{H} = V \sin(\vartheta), \\ \dot{L} = V \cos(\vartheta) \cos(\psi), \\ \dot{Z} = -V \cos(\vartheta) \sin(\psi), \\ \dot{V} = g(n_X - \sin(\vartheta)), \\ \dot{\vartheta} = g \frac{n_Y \cos(\gamma) - \cos(\vartheta)}{V}, \\ \dot{\psi} = -g \frac{n_Y \sin(\gamma)}{V \cos(\vartheta)}. \end{cases} \quad (7)$$

В этой системе уравнений координаты  $(H; L; Z)$  представляют собой координаты положения центра масс БПЛА в нормальной земной системе координат. В качестве управлений движением летательного аппарата рассматриваются перегрузки  $n_X, n_Y$  и угол крена  $\gamma$ . Углы  $\vartheta$  и  $\psi$  задают направление траектории и направление вектора земной скорости  $V$ .

## Постановка задачи

Основной задачей моей научно-исследовательской работы является разработка алгоритма построения программной траектории посадки беспилотного летательного аппарата, обеспечивающего перелет с момента принятия решения о возврате на движущееся прямолинейно и равномерно судно, с учетом обеспечения необходимых ограничений на переменные состояния и управляющие воздействия.

Результатом работы алгоритма является построение условно-оптимальной по времени программной траектории и соответствующих ей программных управлений для посадки БПЛА на судно-носитель, исходя из состояния фазовых векторов БПЛА и и судна в момент принятия решения о посадке (при  $t = 0$ ):

$$\bar{\varphi}_{UAV_0} = (H_{UAV_0}, L_{UAV_0}, Z_{UAV_0}, V_{UAV_0}, \vartheta_{UAV_0}, \psi_{UAV_0}, n_{UAV_{X0}}, n_{UAV_{Y0}}, \gamma_{UAV_0}),$$

$$\bar{\varphi}_{SHIP_0} = (H_{SHIP_0}, L_{SHIP_0}, Z_{SHIP_0}, V_{SHIP_0}, \vartheta_{SHIP_0}, \psi_{SHIP_0}).$$

Предварительно необходимо рассмотреть следующие подзадачи:

1. Решить задачу терминального управления, реализующего к моменту времени  $t = T^*$  приведение БПЛА из заданного начального положения  $\bar{\varphi}_{UAV_0}$  на судно-носитель  $\bar{\varphi}_{SHIP_*}$ ;
2. Нахождение условно-оптимальной по времени программной траектории посадки БПЛА на движущееся судно-носитель и соответствующих ей программных управлений;
3. Создание программного комплекса, обеспечивающего визуализацию результатов.

## Глава 2. Задача терминального управления полетом летательного аппарата

Рассмотрим задачу терминального управления полетом БПЛА [5].

В начальный момент времени  $t = 0$  заданы значения переменных состояния летательного аппарата  $(H_0; L_0; Z_0; V_0; \vartheta_0; \psi_0)$  и значения управлений  $(n_{X0}; n_{Y0}; \gamma_0)$ .

Нужно описать алгоритм управления полетом БПЛА, который к моменту времени  $t = T^*$ , обеспечивает перевод летательного аппарата в терминальное состояние  $(H_*; L_*; Z_*; V_*; \vartheta_*; \psi_*)$  со следующими значениями управлений  $(n_{X*}; n_{Y*}; \gamma_*)$ .

Допустим, что время  $T^*$  известно заранее. Заданные состояния летательного аппарата называются граничными условиями задачи терминального управления.

### 2.1. Решение задачи терминального управления

Для того, чтобы получить траекторию полета, необходимо определить функции  $L$ ,  $H$  и  $Z$ , которыми описывается траектория движения БПЛА. Воспользуемся обратными методами динамики.

Чтобы получить функции управления введём новые управления:  $U_1 = n_X$ ,  $U_2 = n_Y \cos(\gamma)$ ,  $U_3 = n_Y \sin(\gamma)$ . Теперь уравнения математической модели движения летательного аппарата линейны по новым управлениям:

$$\begin{cases} \dot{H} = V \sin(\vartheta), \\ \dot{L} = V \cos(\vartheta) \cos(\psi), \\ \dot{Z} = -V \cos(\vartheta) \sin(\psi), \\ \dot{V} = g(-\sin(\vartheta) + U_1), \\ \dot{\vartheta} = g \left( -\frac{\cos(\vartheta)}{V} + \frac{1}{V} U_2 \right), \\ \dot{\psi} = -\frac{g}{V \cos(\vartheta)} U_3. \end{cases} \quad (8)$$

Затем, для получения управлений, приведём систему (8) к канониче-

скому виду, при помощи введения новых переменных состояния:

$$\begin{cases} x_1 = H, \\ x_2 = L, \\ x_3 = Z, \\ x_4 = V \text{Sin}(\vartheta), \\ x_5 = V \text{Cos}(\vartheta) \text{Cos}(\psi), \\ x_6 = -V \text{Cos}(\vartheta) \text{Sin}(\psi). \end{cases} \quad (9)$$

Можно заметить, что после замены выполняются соотношения  $\dot{x}_1 = x_4, \dot{x}_2 = x_5, \dot{x}_3 = x_6$ .

Вместе с ограничениями  $\{|\vartheta| < \frac{\pi}{2}; |\psi| < \pi; V > 0\}$ , новые переменные определяют гладкую и невырожденную замену переменных, поскольку старые переменные состояния можно выразить по формулам:

$$\begin{cases} H = x_1, \\ L = x_2, \\ Z = x_3, \\ V = \sqrt{x_4^2 + x_5^2 + x_6^2}, \\ \text{Sin}(\vartheta) = \frac{x_4}{\sqrt{x_4^2 + x_5^2 + x_6^2}}, \\ \text{Sin}(\psi) = -\frac{x_6}{\sqrt{x_5^2 + x_6^2}}, \\ \text{Cos}(\psi) = \frac{x_5}{\sqrt{x_5^2 + x_6^2}}. \end{cases} \quad (10)$$

Следовательно, получим, в новых переменных, систему дифференциальных уравнений в каноническом виде:

$$\begin{cases} \dot{x}_1 = x_4, \quad \dot{x}_2 = x_5, \quad \dot{x}_3 = x_6, \\ \dot{x}_4 = -g + u_1 g \text{Sin}(\vartheta) + u_2 \text{Cos}(\vartheta), \\ \dot{x}_5 = u_1 g \text{Cos}(\vartheta) \text{Cos}(\psi) - u_2 g \text{Sin}(\vartheta) \text{Cos}(\psi) + u_3 g \text{Sin}(\psi), \\ \dot{x}_6 = -u_1 g \text{Cos}(\vartheta) \text{Sin}(\psi) + u_2 g \text{Sin}(\vartheta) \text{Sin}(\psi) + u_3 g \text{Cos}(\psi). \end{cases} \quad (11)$$

Выразив  $x_4, x_5, x_6$  через  $x_1, x_2, x_3$ , получим разрешимую, относитель-

но новых управлений, систему из трёх дифференциальных уравнений:

$$\begin{cases} u_1 = \frac{(\ddot{x}_1+g)\text{Sin}(\vartheta)+\ddot{x}_2\text{Cos}(\vartheta)\text{Cos}(\psi)-\ddot{x}_3\text{Cos}(\vartheta)\text{Sin}(\psi)}{g}, \\ u_2 = \frac{(\ddot{x}_1+g)\text{Cos}(\vartheta)-\ddot{x}_2\text{Sin}(\vartheta)\text{Cos}(\psi)+\ddot{x}_3\text{Sin}(\vartheta)\text{Sin}(\psi)}{g}, \\ u_3 = \frac{\ddot{x}_2\text{Sin}(\psi)+\ddot{x}_3\text{Cos}(\psi)}{g}. \end{cases} \quad (12)$$

Можно сделать вывод, что для любой, заданной функциями от времени  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$ , траектории, будут определены управления реализующие эту траекторию. Но, поскольку задача состоит в поиске траектории, удовлетворяющей граничным условиям, необходимо показать, в каких случаях функции  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  будут удовлетворять граничным условиям.

Граничные условия для переменных состояния и управлений определяют граничные значения функций  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  на концах промежутка времени  $[0; T^*]$ , а также их первые и вторые производные.

Граничные условия для введенных переменных состояния и управлений можно получить, подставив начальные граничные значения в формулы, по которым происходила замена переменных.

Решением терминальной задачи, будут являться гладкие функции, удовлетворяющие полученным граничным условиям.

# Глава 3. Построение условно-оптимальной траектории посадки летательного на движущееся судно

## 3.1. Построение пространственной траектории летательного аппарата

Для траектории, заданной гладкими функциями  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  и удовлетворяющей условиям, полученным из граничных условий для переменных состояния, обязательно найдутся реализующие её управления.

Такие функции можно найти среди полиномов пятой степени, поскольку на каждую функцию накладывается по 6 ограничений. В случае полиномов степени меньше 6, система получится замкнутой и не получится подобрать полином, удовлетворяющий заданным ограничениям. В случае же полиномов степени больше 6, система будет избыточной и не получится явно задать коэффициенты полинома.

Допустим, что  $f(t)$  - это полином пятой степени. Для него должны быть выполнены ограничения:

$$\begin{cases} f(0) = x_0, \\ \dot{f}(0) = \dot{x}_0, \\ \ddot{f}(0) = \ddot{x}_0, \\ f(T^*) = x_*, \\ \dot{f}(T^*) = \dot{x}_*, \\ \ddot{f}(T^*) = \ddot{x}_*. \end{cases} \quad (13)$$

Перепишем полином  $f(t)$  в виде:

$$x_i(t) = \sum_{j=0}^2 \frac{x_{i0}^{(j)}}{j!} t^j + \sum_{j=1}^3 C_{ij} t^{2+j}. \quad (14)$$

При любых значениях параметра  $C_{ij}$ , полином (14) будет удовлетворять граничным условиям при  $t = 0$ . Чтобы граничные условия выполня-

лись и при  $t = T^*$ , параметры  $C_{ij}$  должны удовлетворять системе, полученной из соотношений условий, накладываемых на функцию в конечной точке траектории при  $t = T^*$ :

$$\begin{cases} (T^*)^3 C_{i0} + (T^*)^4 C_{i1} + (T^*)^5 C_{i2} = x_{i*} - x_{i0} - \dot{x}_{i0} T^* - 0.5 \ddot{x}_{i0} (T^*)^2, \\ 3(T^*)^2 C_{i0} + 4(T^*)^3 C_{i1} + 5(T^*)^4 C_{i2} = \dot{x}_{i*} - \dot{x}_{i0} - \ddot{x}_{i0} T^*, \\ 6T^* C_{i0} + 12(T^*)^2 C_{i1} + 20(T^*)^3 C_{i2} = \ddot{x}_{i*} - \ddot{x}_{i0}, \end{cases} \quad i = \overline{1, 3}. \quad (15)$$

Поскольку  $T^* \neq 0$ , можно сказать, что определитель матрицы системы линейных алгебраических уравнений равен  $2(T^*)^9$ , и значит, решение этой системы линейных алгебраических уравнений существует, причём единственное. Соответственно, найдя решение, можно будет определить функции  $L(t)$ ,  $H(t)$  и  $Z(t)$ , определяющие пространственную траекторию движения летательного аппарата, удовлетворяющую начальным и конечным условиям задачи терминального управления.

### 3.2. Поиск условно-оптимальной траектории летательного аппарата

С помощью полиномов пятой степени можно получить пространственную траекторию БПЛА, удовлетворяющую наложенным на неё начальным и конечным условиям на промежутке времени  $t$ . Время  $t$  - это единственный параметр, который характеризует вид полиномов, описывающих траекторию движения в данной задаче. На переменные состояния и параметры управления летательного аппарата накладываются ограничения, поскольку очевидно, что любой летательный аппарат имеет максимальную и минимальную скорость, ограничения по высоте полета и др. В связи с этим, введём дополнительные ограничения на переменные состояния и управления:

$$\begin{cases} H_{min} \leq H \leq H_{max}; L_{min} \leq L \leq L_{max}; Z_{min} \leq Z \leq Z_{max}; \\ V_{min} \leq V \leq V_{max}; \vartheta_{min} \leq \vartheta \leq \vartheta_{max}; \psi_{min} \leq \psi \leq \psi_{max}; \\ n_{Xmin} \leq n_X \leq n_{Xmax}; n_{Ymin} \leq n_Y \leq n_{Ymax}; \gamma_{min} \leq \gamma \leq \gamma_{max}. \end{cases} \quad (16)$$

Поиск минимальной по времени движения БПЛА траектории с заданной точностью  $\tau > 0$  заключается в переборе по времени  $t$  траекторий, увеличивая время манёвра  $t$  на величину  $\delta > 0$  (причём  $\delta \geq \tau$ ).

Начальным значением для  $t$  будем считать минимальное время, за которое можно перейти из начального состояния в конечное, двигаясь прямолинейно с максимально допустимой скоростью:

$$t_0 = \frac{\sqrt{(H_* - H_0)^2 + (L_* - L_0)^2 + (Z_* - Z_0)^2}}{V_{max}}. \quad (17)$$

Значением для окончания перебора можно считать  $t_{max} = (t_0 + 5) * 20$ .

Алгоритм для поиска условно-оптимальной по времени траектории движения БПЛА:

- 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  $\delta$ , а потом уменьшить значение  $\delta$  в 2 раза (проводить уменьшение, пока  $\delta$  не станет равным  $\tau$ ).
- Если была найдена траектория, не удовлетворяющая наложенным на неё ограничениям, то значение  $t$  необходимо увеличить на соответствующую величину  $\delta$ .

Первая траектория, которая будет удовлетворять ограничениям при  $\delta = \tau$ , и будет условно-оптимальной по времени с точностью  $\tau$ .

### **3.3. Определение положения судна в момент посадки**

Поскольку судно-носитель движется, то конечное положение БПЛА в момент посадки  $t = T^*$  неизвестно. Для его дальнейшего определения,

предлагается рассмотреть случай равномерного движения судна. В начальный момент времени  $t = t_0$  положения БПЛА и судна описываются фазовыми векторами состояния:

$$\bar{\varphi}_{UAV_0} = (H_{UAV_0}, L_{UAV_0}, Z_{UAV_0}, V_{UAV_0}, \vartheta_{UAV_0}, \psi_{UAV_0}, n_{UAV_{X0}}, n_{UAV_{Y0}}, \gamma_{UAV_0}), \quad (18)$$

$$\bar{\varphi}_{SHIP_0} = (H_{SHIP_0}, L_{SHIP_0}, Z_{SHIP_0}, V_{SHIP_0}, \vartheta_{SHIP_0}, \psi_{SHIP_0}). \quad (19)$$

Так как движение считается равномерным и прямолинейным, то положение судна в любой момент времени  $t = T$  можно описать вектором:

$$\bar{\varphi}_{SHIP_T} = (H_{SHIP_0}, L_{SHIP_T}, Z_{SHIP_T}, V_{SHIP_0}, \vartheta_{SHIP_0}, \psi_{SHIP_0}), \quad (20)$$

где

$$L_{SHIP_T} = L_{SHIP_0} + TV_{SHIP_0} \cos(\psi_{SHIP_0}),$$

$$Z_{SHIP_T} = Z_{SHIP_0} + TV_{SHIP_0} \sin(\psi_{SHIP_0}).$$

Следовательно, зная положение судна, мы можем определить конечное положение БПЛА в момент посадки:

$$\bar{\varphi}_{UAV_{T^*}} = (H_{SHIP_0}, L_{SHIP_{T^*}}, Z_{SHIP_{T^*}}, V_{SHIP_0}, \vartheta_{SHIP_0}, \psi_{SHIP_0}, n_{UAV_{XT^*}}, n_{UAV_{YT^*}}, \gamma_{UAV_{T^*}}). \quad (21)$$

### 3.4. Поиск условно-оптимальной траектории посадки летательного на движущееся судно

Сложность поиска траектории посадки заключается главным образом в подборе времени  $T^*$ , поскольку за время пока БПЛА будет перемещаться из начального положения в точку посадки, судно успеет переместиться на некоторое расстояние.

В связи с этим предлагается следующий алгоритм поиска  $T^*$ :

1. Строим условно-оптимальную траекторию посадки из начального положения БПЛА в начальное положение судна и получаем время  $t_1$  необходимое для такого перехода;
2. Находим новое положение судна, в котором оно окажется за время  $t_1$ ;

3. Строим условно-оптимальную траекторию посадки из начального положения БПЛА в новое положение судна и получаем время  $t_2$  необходимое для такого перехода;
4. Рассчитываем расстояние  $d$  между положением судна в моменты времени  $t_1$  и  $t_2$  и сравниваем его с допустимым значением смещения судна  $\epsilon$ ;
5. Повторяем шаги 2–4 до тех пор, пока смещение положения судна в момент времени  $t_i$  не будет удовлетворять  $d \leq \epsilon$ .

В результате, найденное время  $t_i$  и будет временем  $T^*$ , необходимым для перехода по условно-оптимальной траектории посадки БПЛА на движущееся судно-носитель.

# Результаты тестирования программного комплекса

Для тестирования полученного программного комплекса, ограничениями для переменных состояния и управления были взяты характеристики БПЛА Орлан30 [7]. Были использованы ограничения на скорость, высоту, угол крена и перегрузки. Ограничения для углов курса и наклона траектории были подобраны так, чтобы была выполнена гладкая замена переменных  $\{|\vartheta| < \frac{\pi}{2}; |\psi| < \pi; V > 0\}$ . Ограничения дальности полёта и бокового отклонения подбирались таким образом, чтобы БПЛА имел возможность выполнения сложных и длинных манёвров посадки.

В итоге были получены следующие ограничения:

$$0 \leq H \leq 5000; -10000 \leq L \leq 10000; -10000 \leq Z \leq 10000;$$

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

$$-3 \leq n_X \leq 3; 0.2 \leq n_Y \leq 6; -60 \leq \gamma \leq 60.$$

**Пример 1.** Значения переменных состояния и управлений в момент принятия решения о возврате на судно (при  $t = 0$ ):

$$\bar{\varphi}_{UAV_0} = (500, 0, 0, 160, 0, 0, 0, 1, 0),$$

$$\bar{\varphi}_{SHIP_0} = (5, 800, 800, 80, 0, 0).$$

Время маневра посадки - 62.7072 сек.

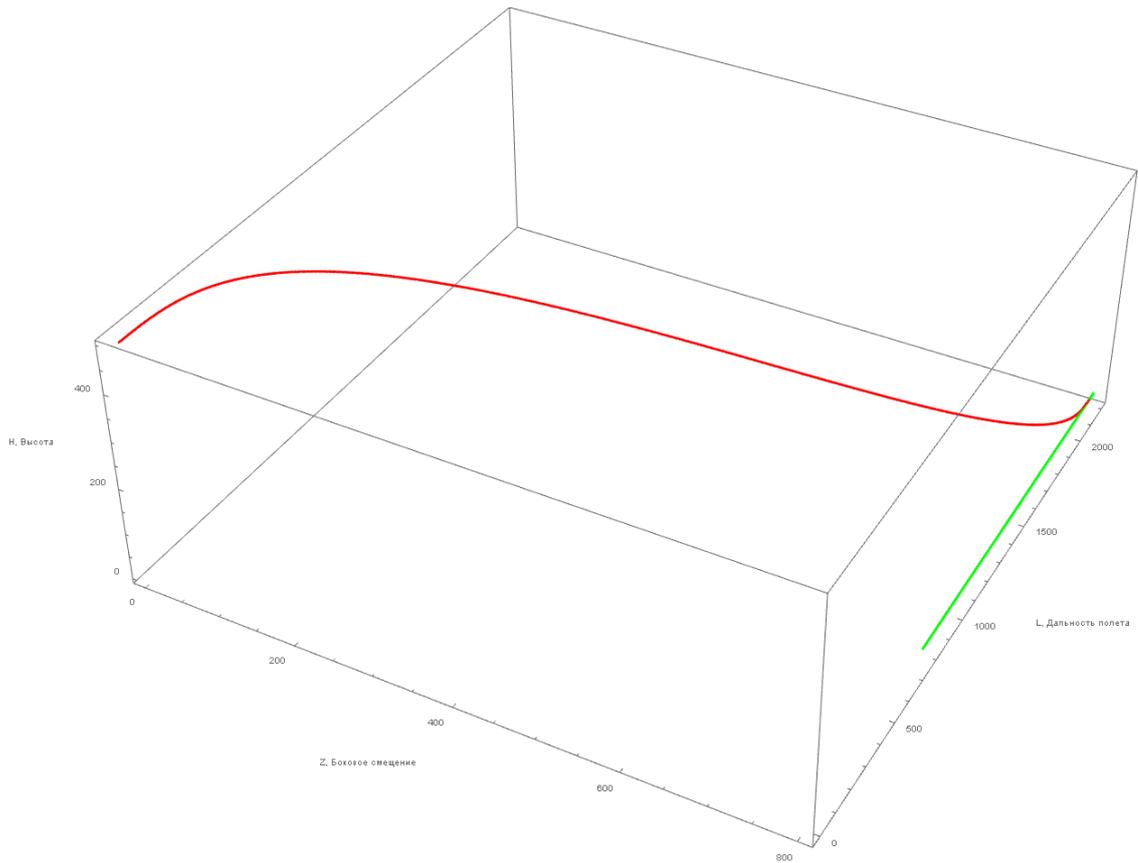


Рис. 3: Траектория посадки БПЛА на судно-носитель в пространстве.

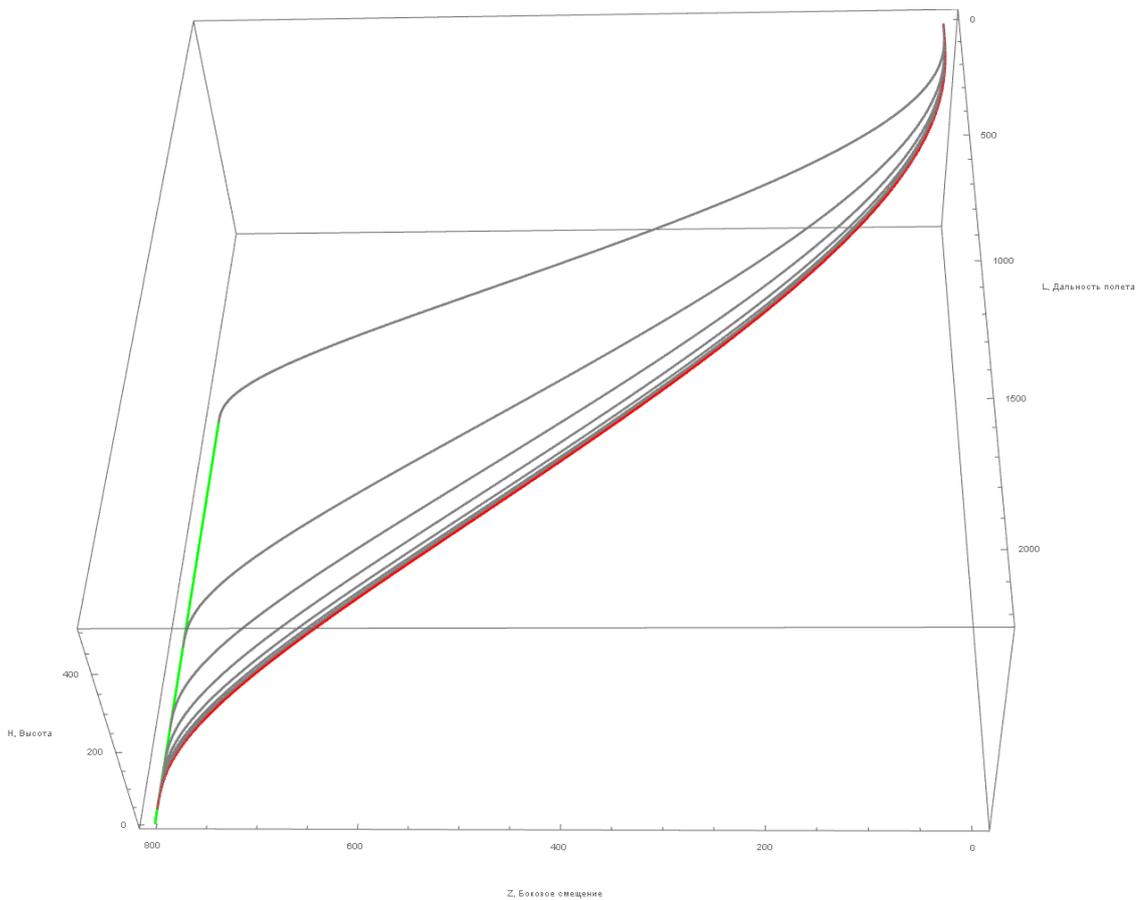


Рис. 4: Траектории в процессе поиска  $T^*$

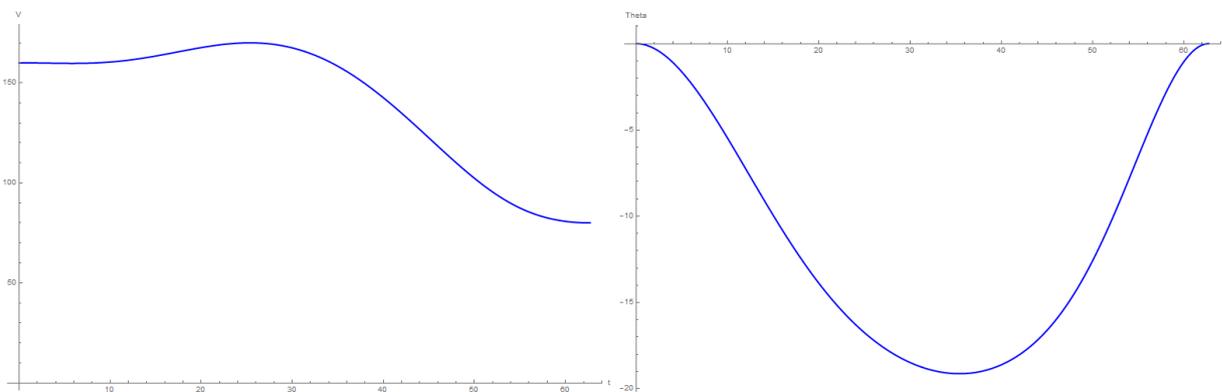


Рис. 5: Графики изменения скорости БПЛА и угла наклона траектории.

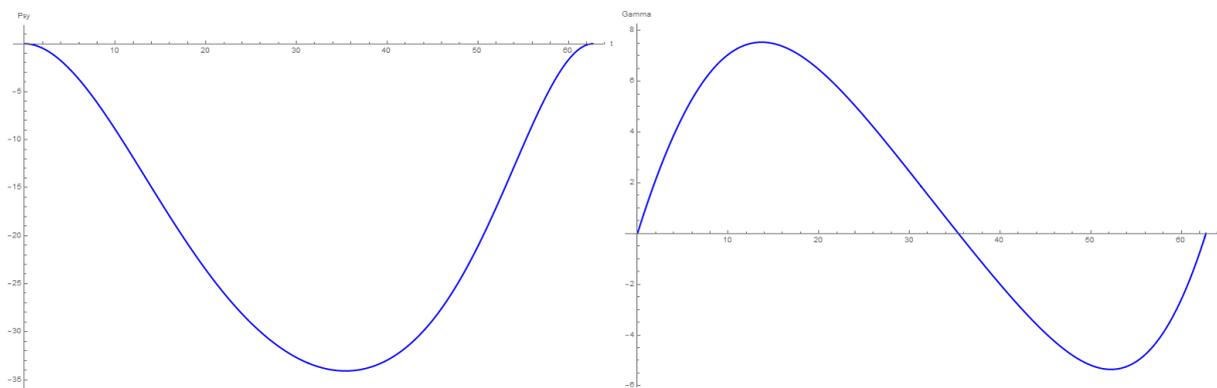


Рис. 6: Графики изменения курсового угла и угла крена.

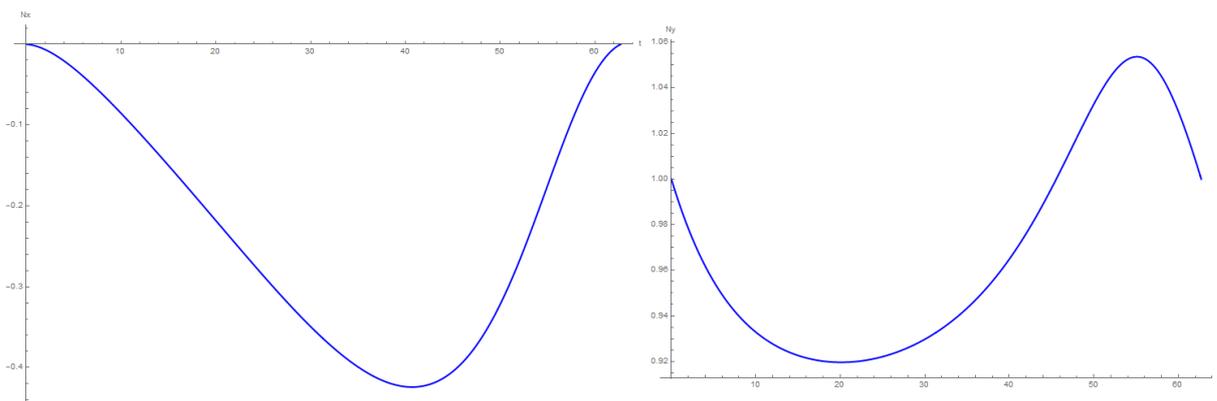


Рис. 7: Графики изменения продольной и поперечной перегрузок.

**Пример 2.** Значения переменных состояния и управлений в момент принятия решения о возврате на судно (при  $t = 0$ ):

$$\bar{\varphi}_{UAV_0} = (800, -200, -500, 160, 0, 30, 0, 1, 0),$$

$$\bar{\varphi}_{SHIP_0} = (5, 1000, 700, 80, 0, 0).$$

Время маневра посадки - 114.944 сек.

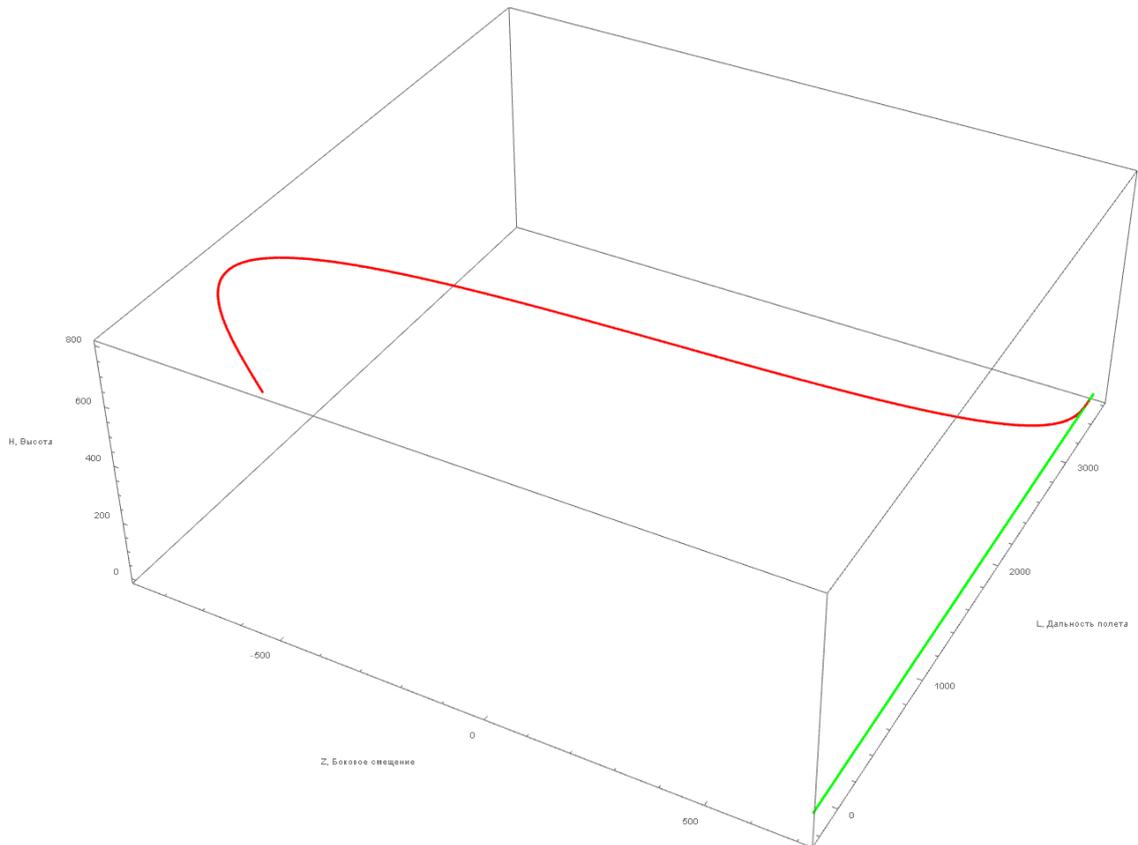


Рис. 8: Траектория посадки БПЛА на судно-носитель в пространстве.

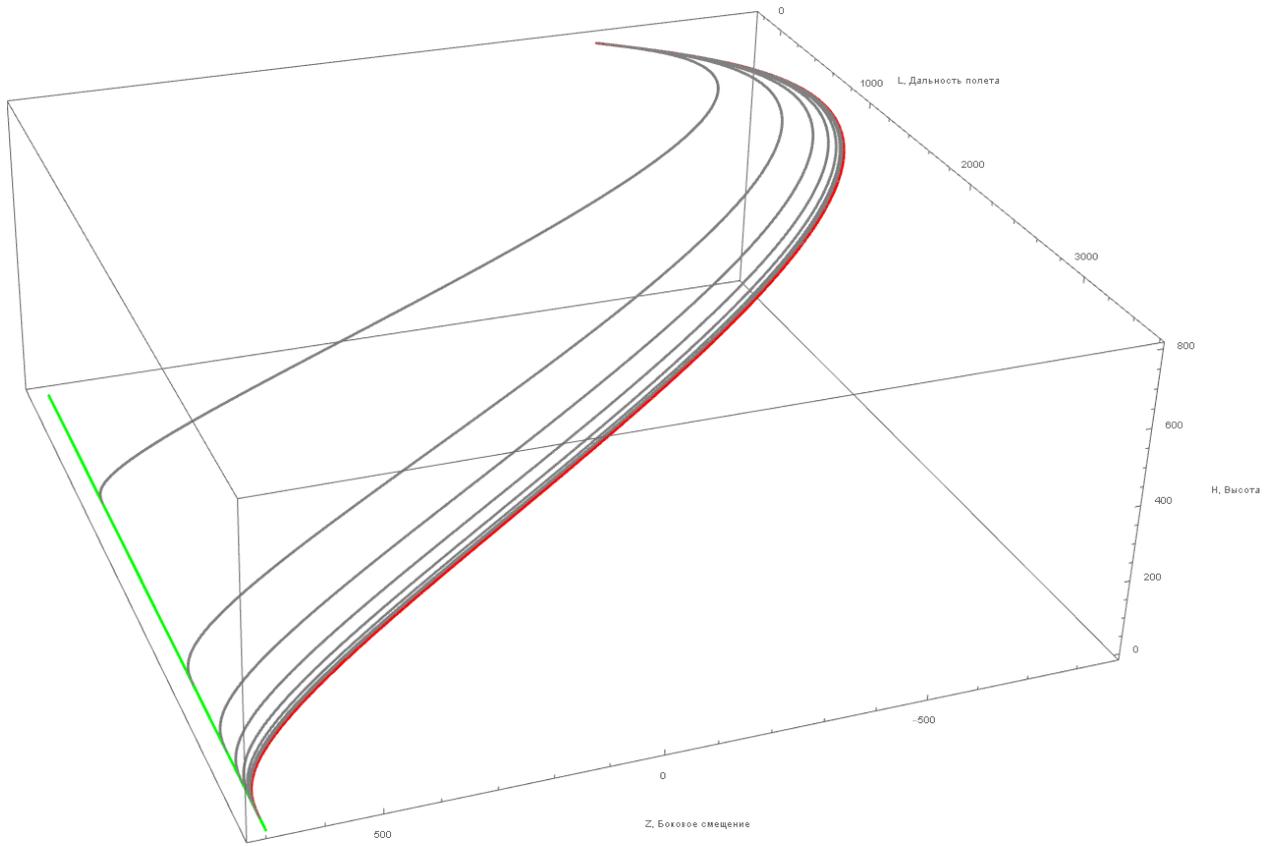


Рис. 9: Траектории в процессе поиска  $T^*$

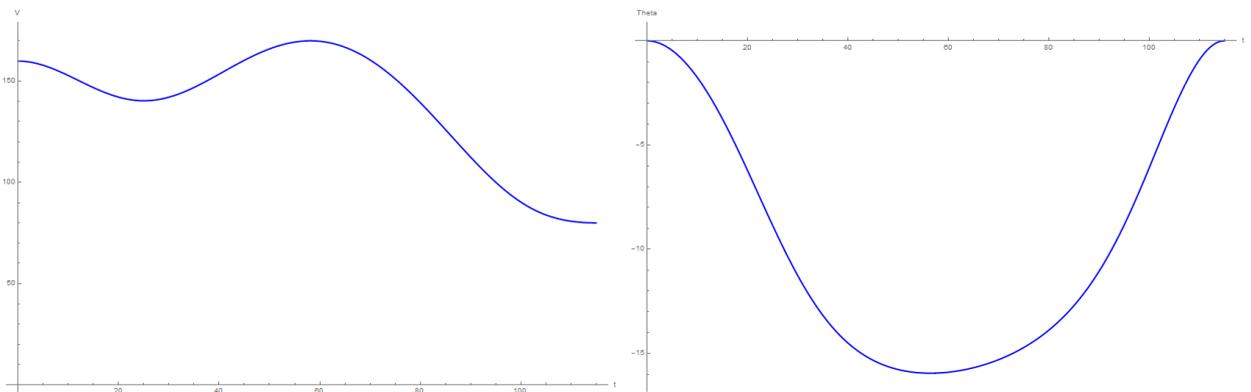


Рис. 10: Графики изменения скорости БПЛА и угла наклона траектории.

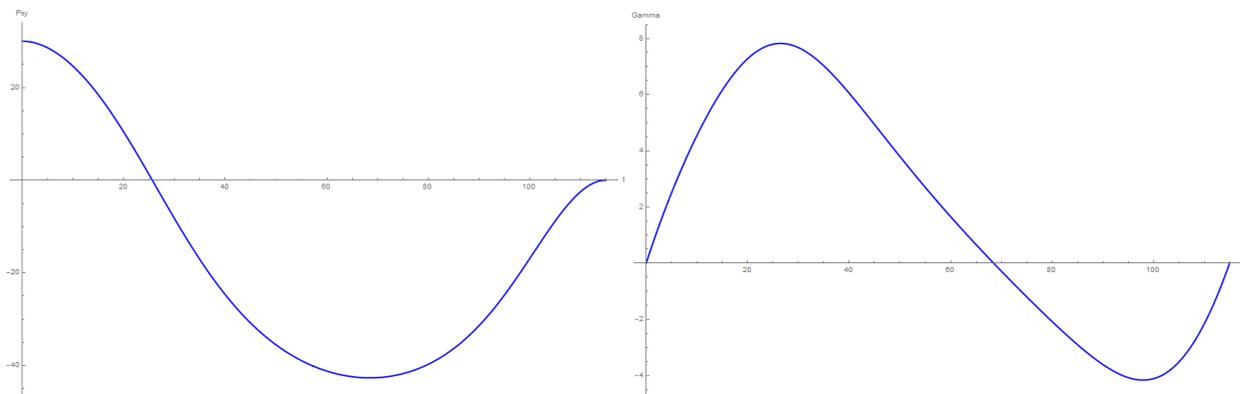


Рис. 11: Графики изменения курсового угла и угла крена.

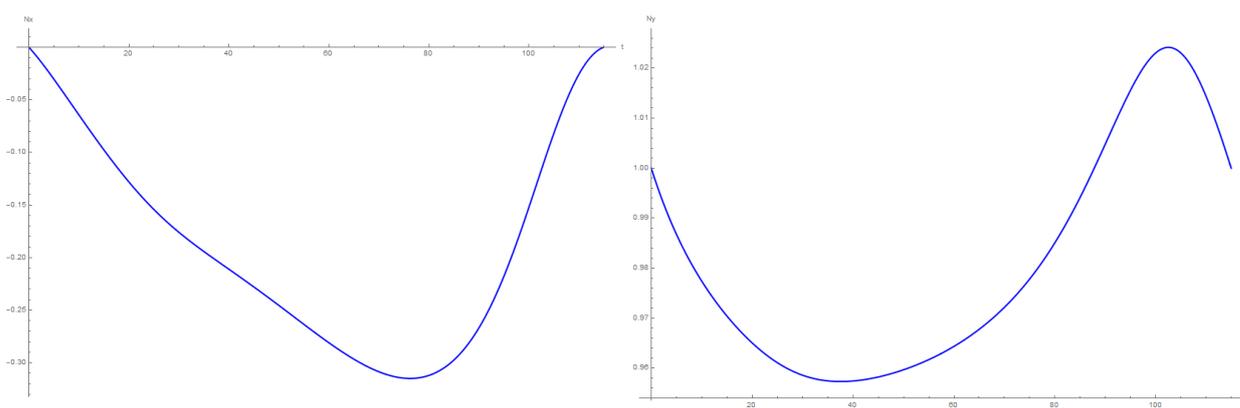


Рис. 12: Графики изменения продольной и поперечной перегрузок.

**Пример 3.** Значения переменных состояния и управлений в момент принятия решения о возврате на судно (при  $t = 0$ ):

$$\bar{\varphi}_{UAV_0} = (1400, 150, 50, 160, 0, 70, 0, 1, 0),$$

$$\bar{\varphi}_{SHIP_0} = (5, 500, 1200, 80, 0, 0).$$

Время маневра посадки - 166.339 сек.

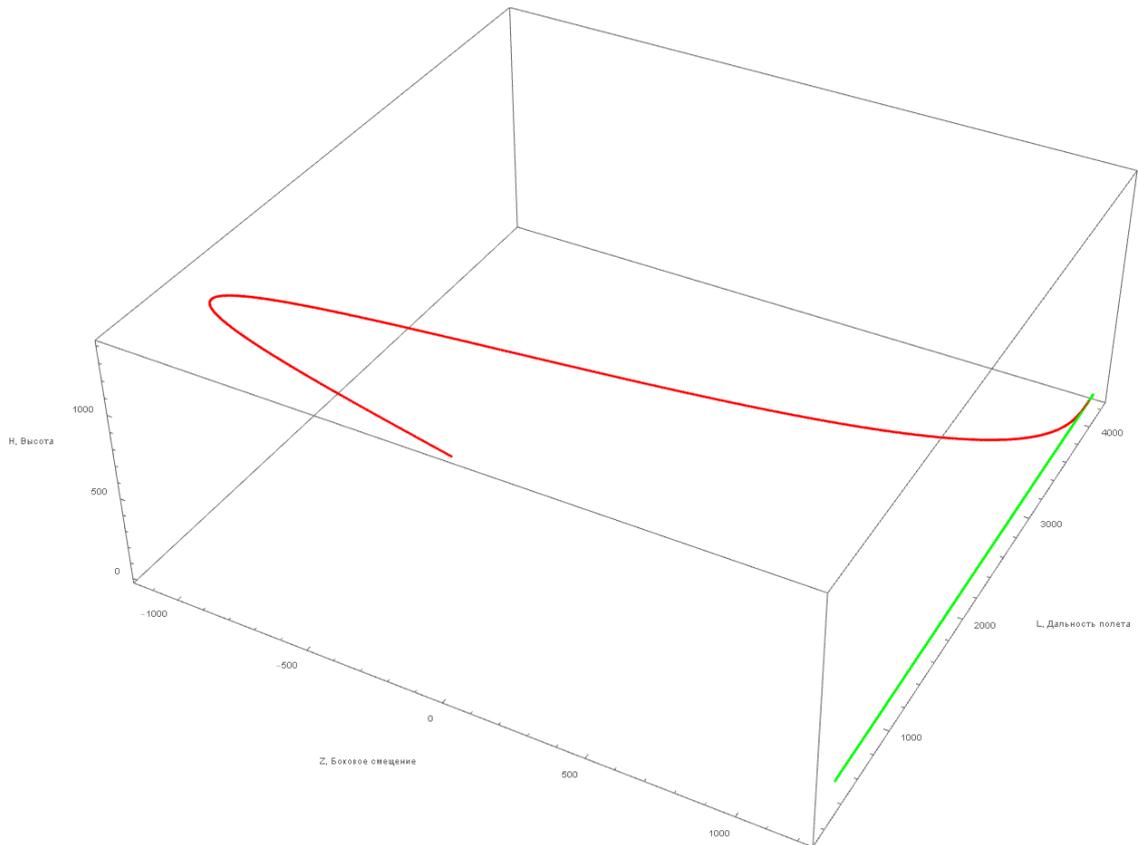


Рис. 13: Траектория посадки БПЛА на судно-носитель в пространстве.

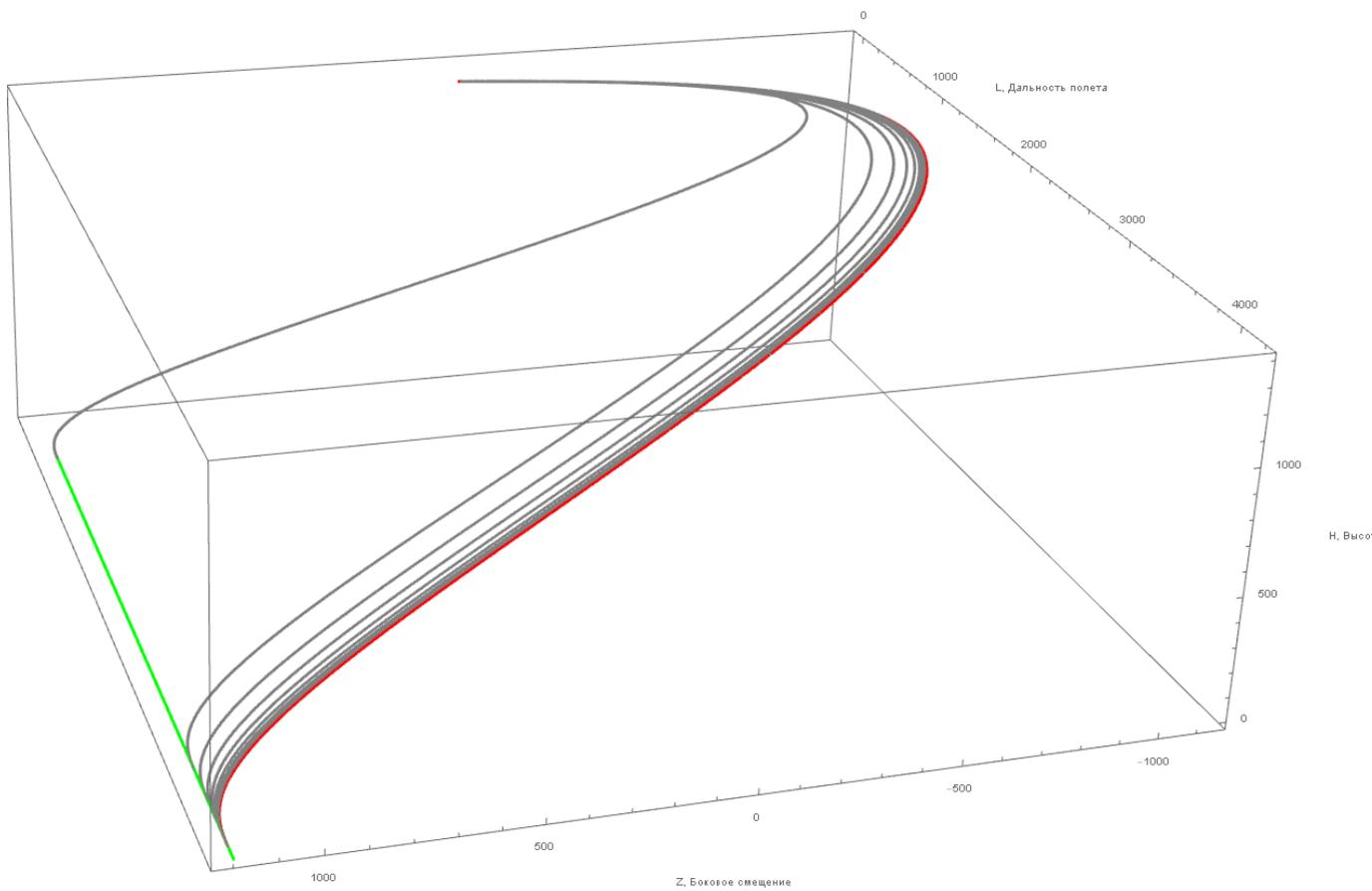


Рис. 14: Траектории в процессе поиска  $T^*$

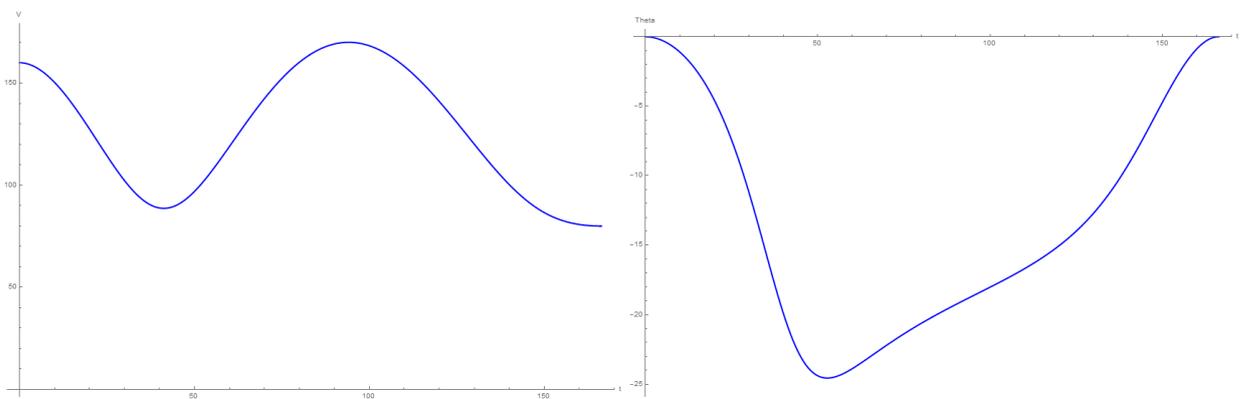


Рис. 15: Графики изменения скорости БПЛА и угла наклона траектории.

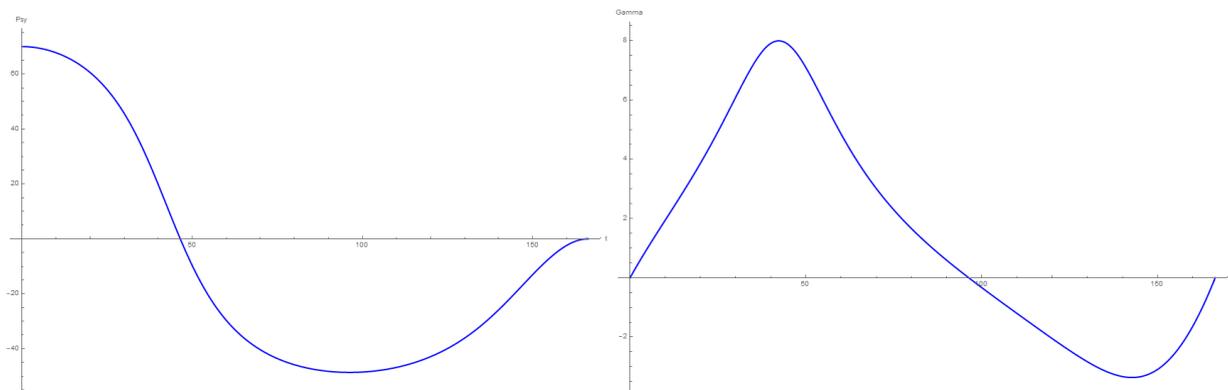


Рис. 16: Графики изменения курсового угла и угла крена.

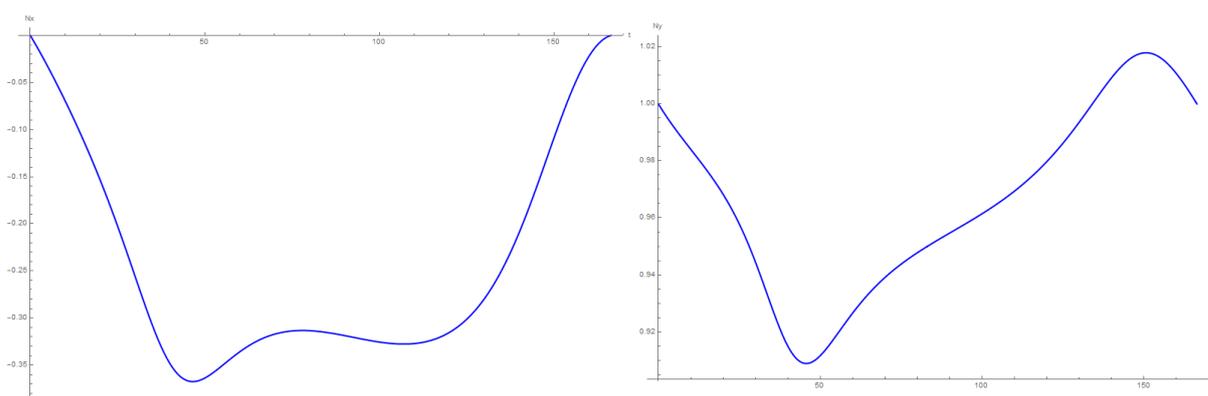


Рис. 17: Графики изменения продольной и поперечной перегрузок.

## Выводы

По итогам проделанной работы:

- В Главе 1 была описана математическая модель движения беспилотного летательного аппарата;
- В Главе 2 было представлено решение задачи терминального управления полетом беспилотного летательного аппарата и приведены уравнения вычисления соответствующих необходимой траектории управлений;
- В Главе 3 был разработан алгоритм построения условно-оптимальная по времени программной траектории посадки беспилотного летательного аппарата на движущееся судно.

Тестирование программного комплекса показало, наиболее частая проблема для реализации более быстрого манёвра посадки - ограничения на скорость БПЛА. Реже реализация траектории упиралась в ограничения летательного аппарата по углу крена и перегрузок.

## Заключение

Была описана математическая модель управляемого движения беспилотного летательного аппарата, решена задача терминального управления движением БПЛА, реализующая его приведение из начального положения на судно-носитель. Также был предложен алгоритм поиска условно-оптимальной по времени программной траектории возврата БПЛА на движущееся судно-носитель.

Был разработан алгоритм построения программной траектории посадки беспилотного летательного аппарата, обеспечивающий перелет с момента принятия решения о возврате на движущееся прямолинейно и равномерно судно, с учетом обеспечения необходимых ограничений на переменные состояния и управляющие воздействия.

Создание программного комплекса и тестирование полученной модели было проведено в математическом пакете Wolfram Mathematica 11.2. Код программы представлен в Приложении.

В дальнейшем, результаты, полученные в данной работе, могут быть использованы для разработки алгоритмов действий БПЛА в случаях потери связи с центром управления полетами, а также для разработки полностью автономных систем управления летательными аппаратами. Рассмотрение влияния качки водной поверхности на траектории возврата БПЛА на судно-носитель, также представляет собой практический интерес.

## Список литературы

1. Крутько П.Д. Обратные задачи динамики управляемых систем. Нелинейные модели. М.: Наука, 1988. — 328 с.
2. Моисеев В.С. Прикладная теория управления беспилотными летательными аппаратами: монография. – Казань: ГБУ «Республиканский центр мониторинга качества образования», 2013. – 768 с.
3. А. Колесников. Новые нелинейные методы управления полетом. М.: Физматлит, 2013. — 196 с.
4. Касаткина Т.С. Преобразования аффинных систем к каноническому виду с использованием замен независимой переменной // Наука и образование, 2013. Вып.7 С. 285-296
5. Канатников А.Н., Шмагина Е.А. Задача терминального управления движением летательного аппарата // Нелинейная динамика и управление, 2010. Вып.7 С. 79-94
6. Онлайн документация к математическому пакету Wolfram Mathematica.  
<http://reference.wolfram.com/language/>
7. СТЦ Орлан30 // Авиационная энциклопедия  
<http://www.airwar.ru/enc/bpla/orlan30.html>

## Приложение

```
ClearAll["Global`*"]
```

```
KOD[] := Module[{},  
  (* Начальное положение *)  
  H0 = SK[[1]]; L0 = SK[[2]]; Z0 = SK[[3]];  
  V0 = SK[[4]]; theta0 = SK[[5]]*Pi/180; psi0 = SK[[6]]*Pi/180;  
  Nx0 = SK[[7]]; Ny0 = SK[[8]]; gamma0 = SK[[9]]*Pi/180;  
  (* Конечное положение *)  
  HFin = FK[[1]]; LFin = FK[[2]]; ZFin = FK[[3]];  
  VFin = FK[[4]]; thetaFin = FK[[5]]*Pi/180; psiFin = FK[[6]]*Pi/180;  
  NxFin = FK[[7]]; NyFin = FK[[8]]; gammaFin = FK[[9]]*Pi/180;  
  trueVector = {};  
  For[i = 0, i <= 1000, i++, trueVector = Append[trueVector, True]];  
  Time0 =  
    Sqrt[(HFin - H0)^2 + (LFin - L0)^2 + (ZFin - Z0)^2]/(VMAX/3.6);  
  TimeFin = (Time0 + 5)*15;  
  deltaTime = 0.5;  
  precision = 0.0001;  
  flagFound = False;  
  flagTimeout = False;  
  curTime = Time0;  
  g = 9.80665;  
  (* Вектор управления в начале *)  
  U0 = {Nx0, Ny0*Cos[gamma0], Ny0*Sin[gamma0]};  
  (* Вектор управления в конце *)  
  UFin = {NxFin, NyFin*Cos[gammaFin], NyFin*Sin[gammaFin]};  
  (* Скорость в метрах в м/с *)  
  V0 = V0/(3.6);  
  VFin = VFin/(3.6);  
  (* Начальное положение в новых координатах *)  
  y10 = H0;  
  y20 = L0;
```

```

y30 = Z0;
y10f = V0*Sin[theta0];
y20f = V0*Cos[theta0]*Cos[psi0];
y30f = -1*V0*Cos[theta0]*Sin[psi0];
y10s = -1*g + g*U0[[1]]*Sin[theta0] + g*U0[[2]]*Cos[theta0];
y20s =
  g*U0[[1]]*Cos[theta0]*Cos[psi0] -
  g*U0[[2]]*Sin[theta0]*Cos[psi0] + g*U0[[3]]*Sin[psi0];
y30s = -1*g*U0[[1]]*Cos[theta0]*Sin[psi0] +
  g*U0[[2]]*Sin[theta0]*Sin[psi0] + g*U0[[3]]*Cos[psi0];
(* Конечное положение в новых координатах *)
y1F = HFin;
y2F = LFin;
y3F = ZFin;
y1Ff = VFin*Sin[thetaFin];
y2Ff = VFin*Cos[thetaFin]*Cos[psiFin];
y3Ff = -1*VFin*Cos[thetaFin]*Sin[psiFin];
y1Fs = -1*g + g*UFin[[1]]*Sin[thetaFin] + g*UFin[[2]]*Cos[thetaFin];
y2Fs =
  g*UFin[[1]]*Cos[thetaFin]*Cos[psiFin] -
  g*UFin[[2]]*Sin[thetaFin]*Cos[psiFin] + g*UFin[[3]]*Sin[psiFin];
y3Fs = -1*g*UFin[[1]]*Cos[thetaFin]*Sin[psiFin] +
  g*UFin[[2]]*Sin[thetaFin]*Sin[psiFin] + g*UFin[[3]]*Cos[psiFin];
nnn = 0;
For[i = 0, (! flagFound) && (curTime < TimeFin), i++,
  nnn = nnn + 1; time = Range[0, curTime, curTime/1000];
  delta = curTime;
  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 = LinearSolve[A, B1];
C2 = LinearSolve[A, B2]; C3 = LinearSolve[A, B3];
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;
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 = (180/Pi)*ArcSin[SinTheta]; CosTheta = Cos[theta*Pi/180];
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 = ArcTan[CosPsi, SinPsi]*180/Pi;
U1 = ((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;
U2 = ((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*SsinPsi)/g;
U3 = ((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 = U3/U2;
gamma = ArcTan[tanGamma]*180/Pi; CosGamma = Cos[gamma*Pi/180];
Nx = U1; Ny = U2/CosGamma; V = V*3.6;
ASD = {};
For[i = 0, i <= 1001, i++,
    If[(((V[[i]] >= VMIN) && (V[[i]] <= VMAX) && (theta[[i]] >=
        thetaMIN) && (theta[[i]] <= thetaMAX) && (psy[[i]] >=
        psiMIN) && (psy[[i]] <= psiMAX) && (H[[i]] >=
        HMIN) && (H[[i]] <= HMAX) && (L[[i]] >= LMIN) && (L[[i]] <=
        LMAX) && (Z[[i]] >= ZMIN) && (Z[[i]] <=
        ZMAX) && (Nx[[i]] >= NxMIN) && (Nx[[i]] <=
        NxMAX) && (Ny[[i]] >= NyMIN) && (Ny[[i]] <=
        NyMAX) && (gamma[[i]] >= gammaMIN) && (gamma[[i]] <=
        gammaMAX)) == True, ASD = Append[ASD, True],
        ASD = Append[ASD, False]]];
If[ASD == trueVector,
    If[deltaTime >= 2*precision, curTime = curTime - deltaTime;
        deltaTime = deltaTime/2,
        If[deltaTime > precision, curTime = curTime - deltaTime;
            deltaTime = precision, flagFound = True]],
    curTime = curTime + deltaTime];];

If[(curTime > TimeFin), flagTimeout = True];

Print["Время маневра ", curTime];
Print["Траектория найдена ", " ", flagFound];

```

```

Print["Таймаут ", " ", flagTimeout];
];

(*Исходные ограничения*)
HMIN = 0; HMAX = 5000; LMIN = -10000; LMAX = 10000; ZMIN = -10000; \
ZMAX = 10000;
VMIN = 0; VMAX = 170; thetaMIN = -89; thetaMAX = 89; psiMIN = -179; \
psiMAX = 179;
NxMIN = -3; NxMAX = 3; NyMIN = 0.2; NyMAX = 6; gammaMIN = -60; \
gammaMAX = 60;

eps = 0.001; HR = {}; LR = {}; ZR = {};

(*Начальное положение БПЛА и судна*)
A0 = {1400, 150, 50, 160, 0, 70, 0, 1, 0};
K0 = {5, 500, 1200, 80, 0, 0, 0, 1, 0};

(*Конечное положение БПЛА и судна*)
KT = {K0[[
1]], (K0[[2]] + tt*(K0[[4]]/3.6)*Cos[K0[[6]]*Pi/180]), (K0[[3]] +
tt*(K0[[4]]/3.6)*Sin[K0[[6]]*Pi/180]), K0[[4]], K0[[5]], K0[[6]],
0, 1, 0};
AT = {K0[[
1]], (K0[[2]] + tt*(K0[[4]]/3.6)*Cos[K0[[6]]*Pi/180]), (K0[[3]] +
tt*(K0[[4]]/3.6)*Sin[K0[[6]]*Pi/180]), A0[[4]], K0[[5]], K0[[6]],
0, 1, 0};

KTptt = {K0[[
1]], (K0[[2]] + ptt*(K0[[4]]/3.6)*Cos[K0[[6]]*Pi/180]), (K0[[3]] +
ptt*(K0[[4]]/3.6)*Sin[K0[[6]]*Pi/180])};
KTtt = {K0[[
1]], (K0[[2]] + tt*(K0[[4]]/3.6)*Cos[K0[[6]]*Pi/180]), (K0[[3]] +
tt*(K0[[4]]/3.6)*Sin[K0[[6]]*Pi/180])};

```

(\*Первая итерация\*)

```
ptt = 0;  
tt = 0;  
SK = A0;  
FK = KT;  
KOD[];  
HR = Join[HR, H];  
LR = Join[LR, L];  
ZR = Join[ZR, Z];
```

(\*2я итерация\*)

```
ptt = tt;  
tt = curTime;  
SK = A0;  
FK = KT;  
KOD[];  
HR = Join[HR, H];  
LR = Join[LR, L];  
ZR = Join[ZR, Z];  
Print["Смещение судна ",  
  Sqrt[(KTtt[[1]] - KTptt[[1]])^2 + (KTtt[[2]] -  
    KTptt[[2]])^2 + (KTtt[[3]] - KTptt[[3]])^2]];
```

```
For[qw = 3,  
  Sqrt[(KTtt[[1]] - KTptt[[1]])^2 + (KTtt[[2]] -  
    KTptt[[2]])^2 + (KTtt[[3]] - KTptt[[3]])^2] > eps, qw++,  
  Print[qw, "я итерация"];  
  ptt = tt;  
  tt = curTime;  
  SK = A0;  
  FK = KT;  
  KOD[];  
  HR = Join[HR, H];  
  LR = Join[LR, L];
```

```

ZR = Join[ZR, Z];
Print["Смещение судна ",
      Sqrt[(KTtt[[1]] - KTptt[[1]])^2 + (KTtt[[2]] -
        KTptt[[2]])^2 + (KTtt[[3]] - KTptt[[3]])^2]];
];
KOD[];

G1 = ListPointPlot3D[Transpose[{ZR, LR, HR}],
  PlotStyle -> Directive[Gray, Thin], AspectRatio -> Automatic,
  AxesLabel -> {"Z, Боковое смещение", "L, Дальность полета",
    "H, Высота"}];
G2 = ListPointPlot3D[Transpose[{Z, L, H}],
  PlotStyle -> Directive[Red, Thick], AspectRatio -> Automatic,
  AxesLabel -> {"Z, Боковое смещение", "L, Дальность полета",
    "H, Высота"}];
Ktime = Range[0, (curTime + 50), (curTime + 50)/1000];
KTT = {K0[[
  1]] + (Ktime*0), (K0[[2]] + (Ktime)*(K0[[4]]/3.6)*
    Cos[K0[[6]]*Pi/180]), (K0[[3]] + (Ktime)*(K0[[4]]/3.6)*
    Sin[K0[[6]]*Pi/180])};
HK = KTT[[1]];
LK = KTT[[2]];
ZK = KTT[[3]];
G0 = ListPointPlot3D[Transpose[{ZK, LK, HK}],
  PlotStyle -> Directive[Green, Thick], AspectRatio -> Automatic,
  AxesLabel -> {"Z, Боковое смещение", "L, Дальность полета",
    "H, Высота"}];
Show[G1, G2, G0]
Show[G2, G0]
{ListLinePlot[Transpose[{time, V}],
  PlotStyle -> Directive[Blue, Thick], AxesLabel -> {"t", "V"}],
ListLinePlot[Transpose[{time, theta}],
  PlotStyle -> Directive[Blue, Thick], AxesLabel -> {"t", "Theta"}],
ListLinePlot[Transpose[{time, psy}],

```

```
PlotStyle -> Directive[Blue, Thick], AxesLabel -> {"t", "Psy"}],  
ListLinePlot[Transpose[{time, gamma}],  
PlotStyle -> Directive[Blue, Thick], AxesLabel -> {"t", "Gamma"}],  
ListLinePlot[Transpose[{time, Nx}],  
PlotStyle -> Directive[Blue, Thick], AxesLabel -> {"t", "Nx"}],  
ListLinePlot[Transpose[{time, Ny}],  
PlotStyle -> Directive[Blue, Thick], AxesLabel -> {"t", "Ny"}]}
```