

Санкт-Петербургский государственный университет

Факультет прикладной математики – процессов управления
Кафедра механики управляемого движения

Гусаров Артём Алексеевич

Выпускная квалификационная работа

Автономное управление движением БПЛА на расчетной траектории

Направление 010300

Фундаментальная информатика и информационные технологии

Научный руководитель:
к. ф.-м. наук, доцент
Шиманчук Д. В.

Санкт-Петербург
2017

Содержание

Введение	3
1. Постановка задачи	4
2. Обзор литературы	5
3. Описание математической модели	6
4. Постановка задачи терминального управления	11
5. Построение траектории и вычисление программных управлений	12
6. Нахождение оптимальной по времени траектории движения	15
7. Построение управлений в форме обратных связей	18
8. Реализация программного комплекса	19
9. Результаты компьютерного моделирования	20
9.1. Моделирование маневров БПЛА	20
9.1.1. Смена эшелона (снижение ЛА)	20
9.1.2. Смена эшелона (набор высоты ЛА)	23
9.1.3. Разворот БПЛА на 180 градусов	26
9.1.4. Смена бокового смещения	29
Выводы	32
Заключение	34
Список литературы	35
Приложение	36

Введение

В современном мире, в связи с развитием технологий, появилась возможность использовать беспилотные летательные аппараты для решения целого спектра задач, как гражданских, так и военных. Беспилотный летательный аппарат (в дальнейшем БПЛА, либо же БЛА) — летательный аппарат, выполняющий полет без человека (пилота) на борту, управляемый удаленно оператором с поверхности Земли, либо же полностью автоматический,двигающийся по заранее заданным с помощью алгоритмов траекториям.

В настоящее время основным направлением развития БПЛА является повышение автономности работы, а также разработка более совершенных алгоритмов автономного движения. Это позволяет расширить круг выполнимых задач и начать использовать беспилотные летательные аппараты там, где раньше это не представлялось возможным. Так, например, их используют в различных спасательных операциях для мониторинга местности, а также для наблюдения, контроля и патрулирования территории. Кроме того, стало возможным решать прикладные задачи в таких областях, как:

- аэрофото- и видеосъемка (картографирование местности);
- разведывание природных запасов;
- метеоразведка;
- мониторинг различных промышленных объектов.

Во время полета управление БПЛА, как правило, осуществляется автоматически с помощью бортового комплекса навигации и управления, в состав которого входят различные виды антенн, система инерциальных датчиков для определения ориентации и параметров движения БПЛА. Также обычно присутствует приемник спутниковой навигации и система воздушных сигналов (для измерения высоты и воздушной скорости).

1. Постановка задачи

В данной научно-исследовательской работе рассматривается проблема реализации автономного движения БПЛА для прохождения маршрута, под которым подразумевается рассчитанная по исходным данным траектория. Разработка компьютерной модели, задающей управления и находящей оптимальную по времени траекторию, является основной целью этой работы.

Поставленная задача может быть решена путем последовательного выполнения следующих этапов:

- Определение параметров физической модели БПЛА;
- Задание математической модели полёта БПЛА;
- Постановка и решение задачи терминального управления;
- Разработка алгоритма поиска оптимального времени полёта;
- Нахождение программной траектории и необходимых управлений;
- Построение управлений в форме обратных связей;
- Создание программного комплекса для визуализации и анализа результатов;
- Исследование адекватности полученной модели.

2. Обзор литературы

За основу для описания и разработки математической модели была взята информация из книг П. Крутько "Обратная задача динамики управляемых систем" [4] и "Обратные задачи динамики в теории автоматического управления" [5].

При рассмотрении дифференциальных уравнений движения БПЛА, а также во время решения задачи терминального управления использовался материал из статьи А. Канатникова и Е. Шмагиной "Задача терминального управления движением летательного аппарата" [2], а также труды А. Крищенко "Преобразование многомерных аффинных управляемых систем" [3].

Для реализации программного комплекса, визуализации и дальнейшего анализа результатов была использована справочная информация к пакету прикладных программ MATLAB [6]. При тестировании работы программы использовалась информация с официального сайта производителя БПЛА Дозор-600 [7].

3. Описание математической модели

При решении задач автоматического управления движением чаще всего используются алгоритмы, в основе которых лежит концепция обратных задач динамики [4, 5]. Примеры применения этой концепции при решении задач управления ЛА можно найти в [1]. В соответствии с их методологией (обратных задач динамики) [4, 5], терминальное управление движением БЛА следует строить поэтапно. Прежде всего, необходимо построить пространственную траекторию движения центра масс летательного аппарата (в дальнейшем ЛА). Который можно считать материальной точкой, учитывая, что в любой момент времени его ориентация позволит обеспечить необходимую сумму действующих на него сил. В данном случае математическая модель является системой из шести дифференциальных уравнений первого порядка, где под управлениями подразумеваются координаты x и y вектора перегрузки, оказывающей влияние на ЛА. На следующем этапе, с учетом построенной пространственной траектории и вычисленных перегрузок, определяется ориентация ЛА и рассчитываются управления. При этом, важно учитывать аэродинамические ограничения ЛА и его характеристики.

Для построения траектории движения ЛА будут использоваться полиномы от времени. В качестве исходных данных выступают векторы перегрузок, скорости и положения в начальный и конечный моменты времени. При построении пространственной траектории вычисляются соответствующие управления. Программные управления есть не что иное, как функции времени, применение которых подразумевает следование летательным аппаратом выбранной траектории. На их основе вычисляются управления в форме обратной связи, беря во внимание фактическое положение ЛА, что позволяет корректировать различные отклонения от заданной пространственной траектории.

Определим следующие системы координат, которые будем использовать для описания модели и записи уравнений движения БПЛА.

За начало O нормальной земной системы координат (далее НЗСК) $Oxyz$ возьмем некоторую точку земной поверхности. Ось Oy направим

по местной вертикали противоположно вектору силы тяжести, а оси Ox и Oz расположим в горизонтальной плоскости так, чтобы они образовали с осью Oy правую систему координат. Предположим, что для решаемой задачи пространственная ориентация НЗСК остается неизменной, а ее саму принято считать инерциальной для небольших интервалов времени. В рамках данной модели кривизной земной поверхности и вращением земли пренебрегают.

Чтобы получить нормальную систему координат (НСК) O_txyz , переместим начало НЗСК в центр масс ЛА. Данная система координат обозначается соответствующе НЗСК, так как играет вспомогательную роль.

Начало O_t траекторной системы координат $O_tX_tY_tZ_t$ поместим в центр масс ЛА, а ось O_tX_t направим по вектору земной скорости ЛА. Ось O_tY_t расположена в местной веритакальной плоскости, содержащей вектор скорости, и направлена вверх, относительно местной вертикали, перпендикулярно оси O_tX_t . До правой системы координат введенные оси дополняет ось O_tZ_t . С помощью угла пути ψ и угла наклона траектории θ зададим положение траекторной системы координат относительно НСК.

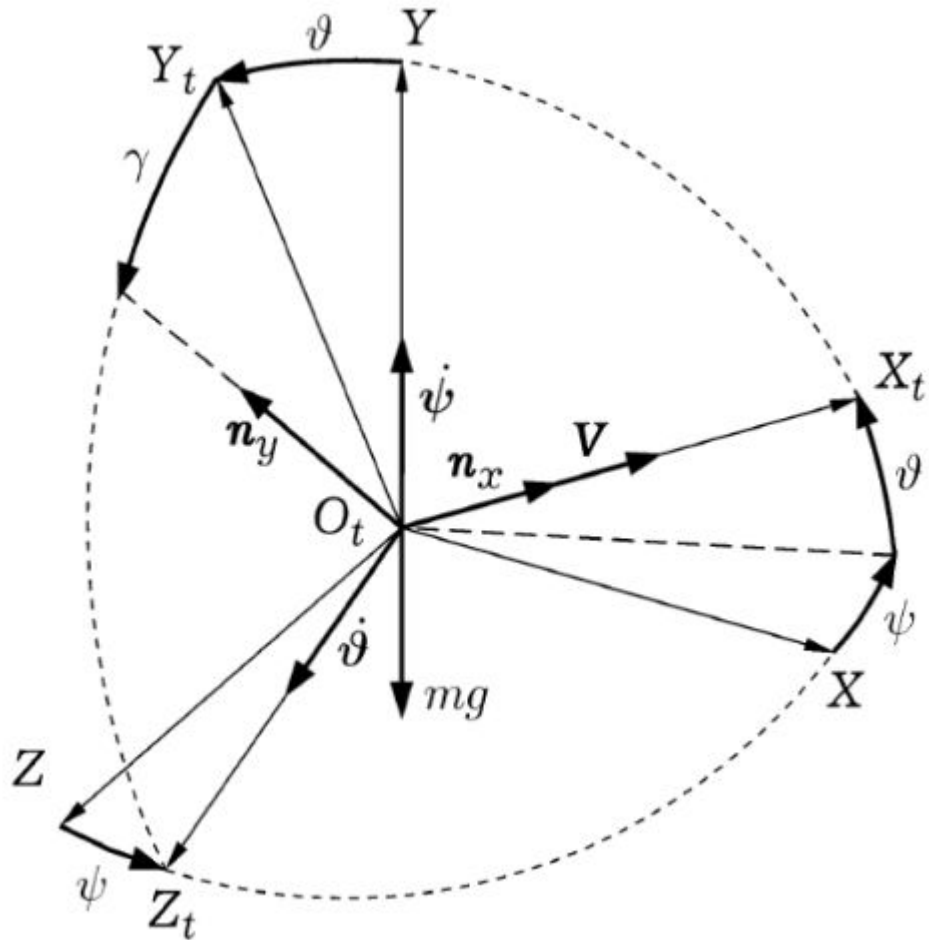


Рис. 1: Траекторная и земная системы координат

Условные обозначения, используемые в данной математической модели и дальнейших вычислениях:

V – скорость ЛА относительно земной поверхности;

θ – угол наклона траектории;

ψ – угол пути;

H – высота полета;

L – дальность полета;

Z – боковая дальность полета;

η_x, η_y – безразмерные перегрузки;

γ – угол крена;

Угол γ и перегрузки η_x, η_y рассматриваются как управления.

Для того, чтобы описать движение центра масс ЛА, запишем следующую систему уравнений [4]:

$$\begin{cases} \dot{V} = (\eta_x - \sin\theta)g, & \dot{H} = V \sin\theta, \\ \dot{\theta} = \frac{(\eta_y \cos\gamma - \cos\theta)g}{V}, & \dot{L} = V \cos\theta \cos\psi, \\ \dot{\psi} = -\frac{\eta_y g \sin\gamma}{V \cos\theta}, & \dot{Z} = -V \cos\theta \sin\psi. \end{cases} \quad (1)$$

Система (1) не является линейной по управлениям, что затрудняет использование концепции обратных задач динамики [3], поэтому перейдем к новым виртуальным управлениям:

$$v_1 = \eta_x, \quad v_2 = \eta_y \cos\gamma, \quad v_3 = \eta_y \sin\gamma. \quad (2)$$

После этого система (1) преобразуется к виду

$$\begin{cases} \dot{V} = -g \sin\theta + g v_1, & \dot{H} = V \sin\theta, \\ \dot{\theta} = -\frac{\cos\theta}{V}g + \frac{g}{V}v_2, & \dot{L} = V \cos\theta \cos\psi, \\ \dot{\psi} = -\frac{g}{V \cos\theta}v_3, & \dot{Z} = -V \cos\theta \sin\psi. \end{cases} \quad (3)$$

Систему (3) можно записать в виде трех уравнений второго порядка

$$\begin{cases} \ddot{y}_1 = -g + v_1 g \sin\theta + v_2 g \cos\theta, \\ \ddot{y}_2 = v_1 g \cos\theta \cos\psi - v_2 g \sin\theta \cos\psi + v_3 g \sin\psi, \\ \ddot{y}_3 = -v_1 g \cos\theta \sin\psi + v_2 g \sin\theta \cos\psi + v_3 g \cos\psi, \end{cases} \quad (4)$$

где

$$\begin{aligned} y_1 &= H, & y_2 &= L, & y_3 &= Z, \\ y_4 &= \dot{y}_1, & y_5 &= \dot{y}_2, & y_6 &= \dot{y}_3, \end{aligned}$$

Следуя следующим формулам, можно говорить об обратимости замены переменных, при соблюдении ограничений:

$$V > 0, \quad |\theta| < \frac{\pi}{2}, \quad -\pi < \psi < \pi.$$

$$\left\{ \begin{array}{l} V = \sqrt{y_4^2 + y_5^2 + y_6^2}, \\ \sin\theta = \frac{y_4}{\sqrt{y_4^2 + y_5^2 + y_6^2}}, \quad \cos\theta = \frac{\sqrt{y_5^2 + y_6^2}}{\sqrt{y_4^2 + y_5^2 + y_6^2}}, \\ \sin\psi = -\frac{y_6}{\sqrt{y_5^2 + y_6^2}}, \quad \cos\psi = \frac{y_5}{\sqrt{y_5^2 + y_6^2}}. \end{array} \right. \quad (5)$$

Таким образом, мы перейдем к постановке задачи терминального управления движением БПЛА, что будет рассмотрено в следующей главе.

4. Постановка задачи терминального управления

Рассмотрим следующую задачу терминального управления. Пусть в начальный момент времени t_0 заданы значения переменных фазового состояния ЛА:

$$\begin{cases} H = H_0, & L = L_0, \\ Z = Z_0, & V = V_0, \\ \theta = \theta_0, & \psi = \psi_0, \end{cases} \quad (6)$$

и значения компонент управления

$$\eta_x = \eta_{x_0}, \quad \eta_y = \eta_{y_0}, \quad \gamma = \gamma_0. \quad (7)$$

Требуется рассчитать оптимальную по времени траекторию и синтезировать алгоритм управления, который к соответствующему моменту времени t_1 обеспечивает перевод ЛА в терминальное состояние

$$\begin{cases} H = H_1, & L = L_1, \\ Z = Z_1, & V = V_1, \\ \theta = \theta_1, & \psi = \psi_1, \end{cases} \quad (8)$$

со следующими значениями управляющих компонент:

$$\eta_x = \eta_{x_1}, \quad \eta_y = \eta_{y_1}, \quad \gamma = \gamma_1. \quad (9)$$

Решение данной задачи найдем с помощью описанной ранее математической модели, записанной в форме (4).

5. Построение траектории и вычисление программных управлений

Для решения поставленной задачи воспользуемся концепцией обратных задач динамики. Первым этапом станет выбор удовлетворяющей заданным условиям (в начальный и конечный моменты времени) траектории. Затем вычислим управления, которые обеспечат движение ЛА по заданной траектории.

Значения функций $y_1(t)$, $y_2(t)$, $y_3(t)$ известны при заданной траектории, а их значения при $t = t_0$ можно определить условиями (6):

$$\begin{aligned} y_1(t) &= H(t), & y_2(t) &= L(t), & y_3(t) &= Z(t) \\ y_{10} &= H_0, & y_{20} &= L_0, & y_{30} &= Z_0. \end{aligned}$$

Запишем производные функций $y_1(t)$, $y_2(t)$, $y_3(t)$ при $t = t_0$, которые определяются начальными значениями переменных состояния (6), согласно уравнениям (3), а также их вторые производные при $t = t_0$, используя начальные значения управлений (7):

$$\left\{ \begin{array}{l} y_{10} = V_0 \sin \theta_0, \\ y_{20} = V_0 \cos \theta_0 \cos \psi_0, \\ y_{30} = -V_0 \cos \theta_0 \sin \psi_0 \\ \ddot{y}_{10} = -g + v_{10} g \sin \theta_0 + v_{20} g \cos \theta_0, \\ \ddot{y}_{20} = v_{10} g \cos \theta_0 \cos \psi_0 - v_{20} g \sin \theta_0 \cos \psi_0 + v_{30} g \sin \psi_0, \\ \ddot{y}_{30} = -v_{10} g \cos \theta_0 \sin \psi_0 + v_{20} g \sin \theta_0 \sin \psi_0 + v_{30} g \cos \psi_0, \end{array} \right.$$

Аналогичным образом найдем значения первых и вторых производных функций $y_1(t)$, $y_2(t)$, $y_3(t)$ при $t = t_1$, согласно (8, 9):

$$\left\{ \begin{array}{l} y_{11} = H_1, y_{21} = L_1, y_{31} = Z_1; \\ \dot{y}_{11} = V_1 \sin \theta_1, \\ y_{21} = V_1 \cos \theta_1 \cos \psi_1, \\ y_{31} = -V_1 \cos \theta_1 \sin \psi_1; \\ \ddot{y}_{11} = -g + v_{11} g \sin \theta_1 + v_{21} g \cos \theta, \\ \ddot{y}_{21} = v_{11} g \cos \theta_1 \cos \psi_1 - v_{21} g \sin \theta_1 \cos \psi_1 + v_{31} g \sin \psi_1, \\ \ddot{y}_{31} = -v_{11} g \cos \theta_1 \sin \psi_1 + v_{21} g \sin \theta_1 \sin \psi_1 + v_{31} g \cos \psi_1. \end{array} \right.$$

Следовательно, траектория движения ЛА, удовлетворяющая условиям (6)–(9), накладывает граничные условия на функции $y_1(t)$, $y_2(t)$, $y_3(t)$, определенные на отрезке $[t_0, t_1]$. Будем находить эти функции в виде полиномов пятой степени, исходя из шести условий, накладываемых на каждую из функций.

Рассмотрим далее задачу построения многочлена $y_i(t)$, $i = \overline{1, 3}$, пятой степени, которую удобно будет свести к случаю отрезка $[0, 1]$, и запишем условия, которым удовлетворяет многочлен:

$$\begin{aligned} y_i(t_0) &= y_{i0}, \quad \dot{y}_i(t_0) = \dot{y}_{i0}, \quad \ddot{y}_i(t_0) = \ddot{y}_{i0}, \\ y_i(t_1) &= y_{i1}, \quad \dot{y}_i(t_1) = \dot{y}_{i1}, \quad \ddot{y}_i(t_1) = \ddot{y}_{i1}. \end{aligned}$$

Пусть $\Delta = t_1 - t_0$, $p_i(\tau) = y_i(t_0 + \tau\Delta)$, тогда условия, которым удовлетворяет многочлен $p_i(\tau)$, выглядят следующим образом [2]:

$$\begin{aligned} p_i(0) &= y_{i0}, \quad p'_i(0) = \dot{y}_{i0}\Delta, \quad p''_i(0) = \ddot{y}_{i0}\Delta^2, \\ p_i(1) &= y_{i1}, \quad p'_i(1) = \dot{y}_{i1}\Delta, \quad p''_i(1) = \ddot{y}_{i1}\Delta^2, \end{aligned}$$

где штрих – дифференцирование по τ .

Перепишав многочлен $p_i(\tau)$ в виде $p_i(\tau) = \alpha_0 + \alpha_1\tau + \dots + \alpha_5\tau^5$ и записав предыдущие соотношения при $\tau = 1$ получим систему из трех

уравнений с тремя неизвестными:

$$\begin{cases} \alpha_3 + \alpha_4 + \alpha_5 = y_{i1} - y_{i0} - \dot{y}_{i0}\Delta - \frac{1}{2}\ddot{y}_{i0}\Delta^2, \\ 3\alpha_3 + 4\alpha_4 + 5\alpha_5 = \dot{y}\Delta_{i1} - \dot{y}_{i0}\Delta - \ddot{y}_{i0}\Delta^2, \\ 6\alpha_3 + 12\alpha_4 + 20\alpha_5 = \ddot{y}_{i1}\Delta^2 - \ddot{y}_{i0}\Delta^2. \end{cases} \quad (10)$$

Так как определитель системы не равен 0, она имеет единственное решение:

$$\begin{cases} \alpha_3 = 10y_{i1} - 10y_{i0} - 6\dot{y}_{i0}\Delta - \frac{3}{2}\ddot{y}_{i0}\Delta^2 - 4\dot{y}_{i1}\Delta + \frac{1}{2}\ddot{y}_{i1}\Delta^2, \\ \alpha_4 = -15y_{i1} + 15y_{i0} + 8\dot{y}_{i0}\Delta + \frac{3}{2}\ddot{y}_{i0}\Delta^2 + 7\dot{y}_{i1}\Delta - \ddot{y}_{i1}\Delta^2, \\ \alpha_5 = 6y_{i1} - 6y_{i0} - 3\dot{y}_{i0}\Delta - \frac{1}{2}\ddot{y}_{i0}\Delta^2 - 3\dot{y}_{i1}\Delta + \frac{1}{2}\ddot{y}_{i1}\Delta^2. \end{cases} \quad (11)$$

Тем самым построена траектория (равносильно построению функций $y_i(t) = p_i(\frac{t-t_0}{\Delta})$, $i = \overline{1, 3}$), тогда программные управления v_1, v_2, v_3 вычисляются следующим образом (решая систему (4) относительно этих управлений) [2]:

$$\begin{cases} v_1 = \sin\theta + \frac{\ddot{y}_1 \sin\theta + \ddot{y}_2 \cos\theta \cos\psi - \ddot{y}_3 \cos\theta \sin\psi}{g}, \\ v_2 = \cos\theta + \frac{\ddot{y}_1 \cos\theta - \ddot{y}_2 \sin\theta \cos\psi + \ddot{y}_3 \sin\theta \sin\psi}{g}, \\ v_3 = \frac{\ddot{y}_3 \cos\psi + \ddot{y}_2 \sin\psi}{g}, \end{cases} \quad (12)$$

где по формулам (5) вычисляются $\cos\theta, \sin\theta, \cos\psi, \sin\psi$, а $y_i, \dot{y}_i, \ddot{y}_i, i = \overline{1, 3}$, уже вычислены. Наконец, находим управления η_x, η_y, γ , используя (2).

6. Нахождение оптимальной по времени траектории движения

Время движения летательного аппарата t оказывает большое влияние на форму траектории, описываемой полиномами. Так, на практике, небольшие изменения времени t могут приводить к существенному изменению поведения переменных состояния и управлений от времени. Принимая это во внимание, найдем его оптимальное значение.

Следует начать с рассмотрения физических рамок возможностей БПЛА, для этого введем ограничения на переменные состояния и управления:

$$\left\{ \begin{array}{l} H_{min} \leq H \leq H_{max}, \\ L_{min} \leq L \leq L_{max}, \\ Z_{min} \leq Z \leq Z_{max}, \\ V_{min} \leq V \leq V_{max}, \\ \theta_{min} \leq \theta \leq \theta_{max}, \\ \psi_{min} \leq \psi \leq \psi_{max}, \\ \eta_{x_{min}} \leq \eta_x \leq \eta_{x_{max}}, \\ \eta_{y_{min}} \leq \eta_y \leq \eta_{y_{max}}, \\ \gamma_{min} \leq \gamma \leq \gamma_{max}. \end{array} \right.$$

Далее, будем увеличивать время маневра летательного аппарата t на некоторую величину δ ($\delta > 0$ и $\delta \geq \epsilon$, где $\epsilon > 0$ – заранее заданная точность), перебирая траектории в порядке возрастания времени t . За начальное значение времени t_0 возьмем минимально возможное время перехода из начального положения в конечное, двигаясь при этом с максимальной скоростью V_{max} и прямолинейно:

$$t_0 = \frac{\sqrt{(H_1 - H_0)^2 + (L_1 - L_0)^2 + (Z_1 - Z_0)^2}}{V_{max}}.$$

В процессе перебора на каждом этапе необходимо проверять управления и реализуемую ими траекторию на соответствие заданным ограничениям. В случае успешного нахождения траектории, удовлетворяющей граничным значениям переменных состояния и управлений, следует уменьшить соответствующее ей значение времени t на величину δ , а значение δ уменьшить в 2 раза. Если же полученная траектория не удовлетворяет наложенным на нее ограничениям, то необходимо увеличить время t на соответствующее значение δ . Таким образом, первая найденная траектория, удовлетворяющая ограничениям при $\delta = \epsilon$ будет являться оптимальной по времени (с заданной точностью ϵ).

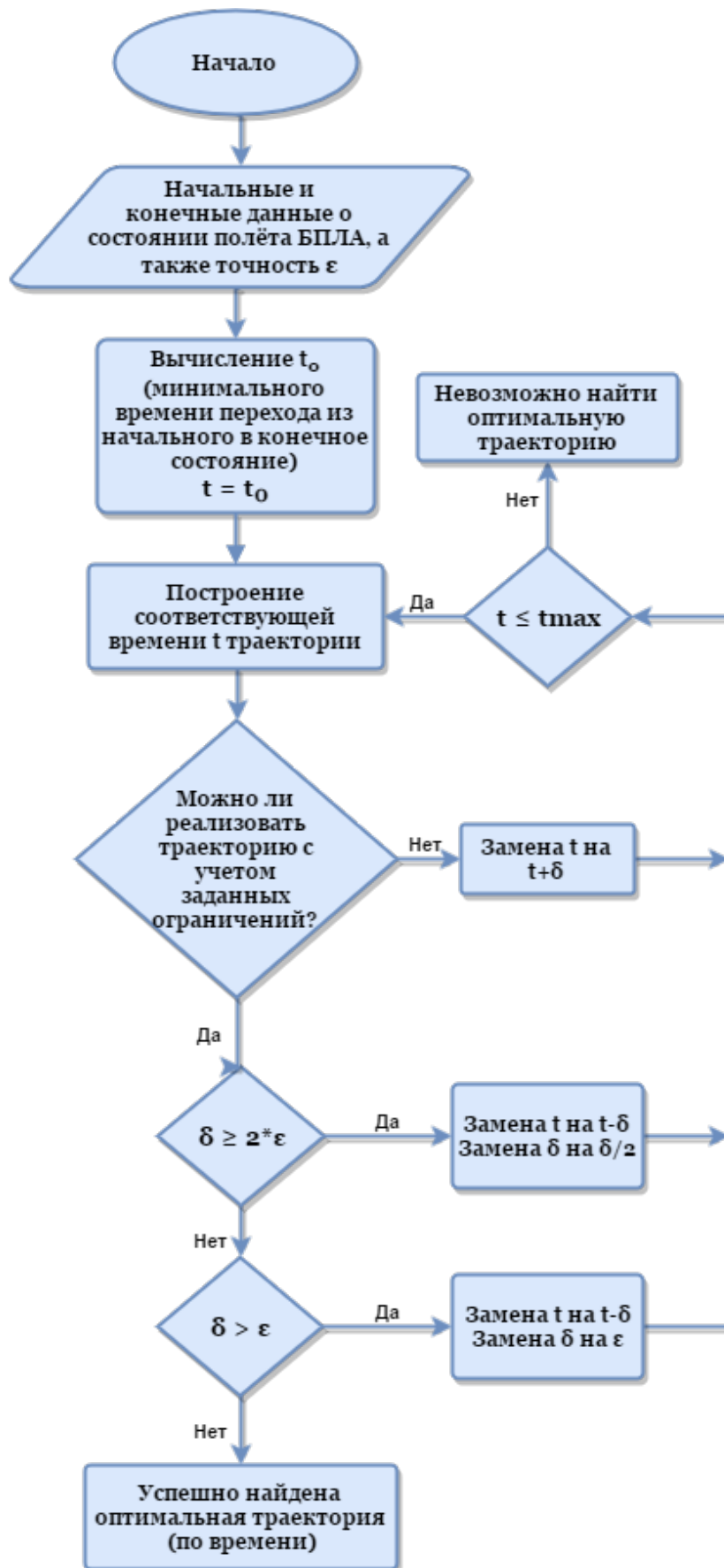


Рис. 2: блок-схема алгоритма (поиск оптимальной траектории)

7. Построение управлений в форме обратных связей

Перепишав систему (4) в векторной форме, получим $\dot{y} = A + B(y, \dot{y})v$, где [2]

$$\left\{ \begin{array}{l} y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} H \\ L \\ Z \end{pmatrix}, v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, A = -g \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \\ B(y, \dot{y}) = g \begin{pmatrix} \sin\theta & \cos\theta & 0 \\ \cos\theta\cos\psi & -\sin\theta\cos\psi & \sin\psi \\ -\cos\theta\sin\psi & \sin\theta\sin\psi & \cos\psi \end{pmatrix}. \end{array} \right. \quad (13)$$

Представим управление в форме обратной связи в виде [2]:

$$v(y, \dot{y}) = \frac{1}{g^2} B^T(y, \dot{y}) B(y_p, \dot{y}_p) v_p - \frac{1}{g^2} B^T(y, \dot{y}) (K_1(\dot{y} - \dot{y}_p) + K_2(y - y_p)), \quad (14)$$

где K_1, K_2 – диагональные матрицы с диагональными элементами k_{i1} и k_{i2} , а y_p и v_p – заданная траектория движения ЛА и программные управления, реализующие ее.

Первым шагом вычислим полиномы $p_i(\tau), i = \overline{1,3}$ (см. п. 5), что означает нахождение пространственной траектории ЛА. Далее, по этим полиномам в текущий момент времени t найдем векторы $y_p(t), \dot{y}_p(t)$, которые определяют состояние, в котором должен находиться ЛА в текущий момент времени, а также вектор программных управлений $v_p(t)$. Учитывая текущее состояние ЛА и формулы (3) находим векторы y и \dot{y} :

$$y = \begin{pmatrix} H \\ L \\ Z \end{pmatrix}, \dot{y} = \begin{pmatrix} \dot{H} \\ \dot{L} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} V \sin\theta \\ V \cos\theta \cos\psi \\ -V \cos\theta \sin\psi \end{pmatrix} \quad (15)$$

Теперь, применяя формулу (14), можно вычислить управление в форме обратной связи.

8. Реализация программного комплекса

Для реализации алгоритма и программного комплекса использовался пакет математического моделирования MATLAB. Основным преимуществом его использования стало наличие большого количества встроенных функций, а также возможность удобной отрисовки результатов вычислений для последующего их анализа.

Результатом работы стала реализация поиска оптимальной по времени траектории ЛА и её моделирование при заданных ограничениях. Кроме того, используя разработанный программный комплекс можно анализировать и сравнивать поведение параметров состояния и управлений, их изменение с течением времени в процессе манёвра и зависимости друг от друга.

Ограничения на переменные состояния и управления, а также их значения в начальном и конечном состоянии траектории БПЛА задаются в соответствующей *.m* функции *trajectory*, как и требуемая точность ϵ . В ней же определяются виртуальные управления, преобразуются переменные и в новом виде описываются условия. Далее, происходит запуск в цикле по времени следующей функции *trajectory2*. Если же во время вычисления оптимальную траекторию не удалось найти, либо же затрачиваемое время принимает слишком большое значение, не удовлетворяющее заданному промежутку, логическая метка, возвращаемая функцией, принимает соответствующее значение.

За выполнение основных расчетов отвечает *.m* функция *trajectory2*. Именно в ней происходит получение численных значений полиномов, задающих траекторию, и управлений, реализующих ее. Проверка найденной траектории на предмет удовлетворения заданным ограничениям также происходит в процессе работы данной *.m* функции. В случае успешного нахождения оптимального значения времени, функция возвращает логическую метку с соответствующим значением.

Итогом работы программы становится готовый для визуализации и анализа набор данных, описывающих оптимальную по времени траекторию маневра ЛА и его состояние в течение этого времени.

9. Результаты компьютерного моделирования

При моделировании все вычисления выполнялись, учитывая физические ограничения новейшего российского разведывательно-ударного БПЛА Дозор-600 [7], взятого в качестве примера для построения оптимальной пространственной траектории и расчета реализующих ее управлений. Для представления возможностей программного комплекса и оценки адекватности полученной модели, были смоделированы такие маневры, как смена эшелона, а именно снижение ЛА и набор высоты, а также разворот на 180 градусов.

	MIN	MAX
V , км/ч	70	210
H , м	250	7500
L , м	-25000	25000
Z , м	-25000	25000
θ , град	-89	89
ψ , град	-179	179
η_x	-3.5	3.5
η_y	0.5	5
γ , град	-60	60

Таблица 1: Граничные условия для дальнейшего моделирования

9.1. Моделирование маневров БПЛА

9.1.1. Смена эшелона (снижение ЛА)

	V (км/ч)	H (м)	L (м)	Z (м)	θ (град)	ψ (град)	γ (град)	η_x	η_y
при t_0	130	2500	0	0	-20	0	0	0	1
при t_{opt}	110	500	2300	700	0	0	0	0	1

Таблица 2: Исходные данные для выполнения маневра

При моделировании оптимальной по времени траектории смены эшелона, получили $t_{opt} = 73.182$, при заданной точности $\epsilon = 0.001$. Далее демонстрируется непосредственно сама траектория движения ЛА при полученном оптимальном значении времени, а также графики изменения переменных состояния и управлений с течением времени.

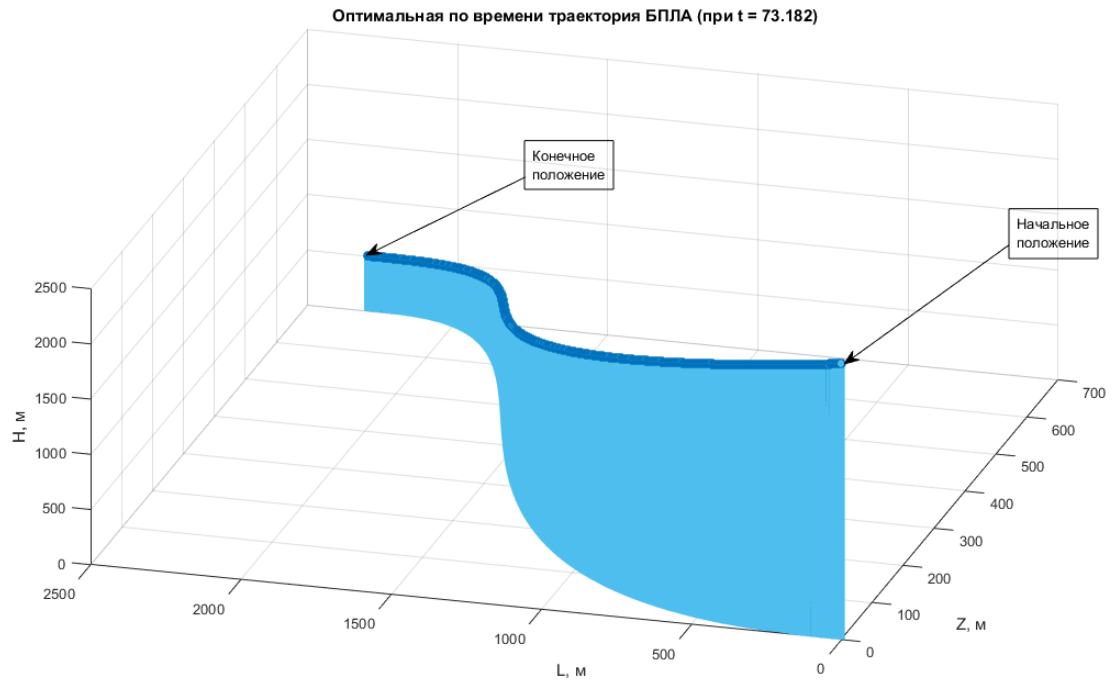


Рис. 3: Оптимальная по времени t траектория движения ЛА

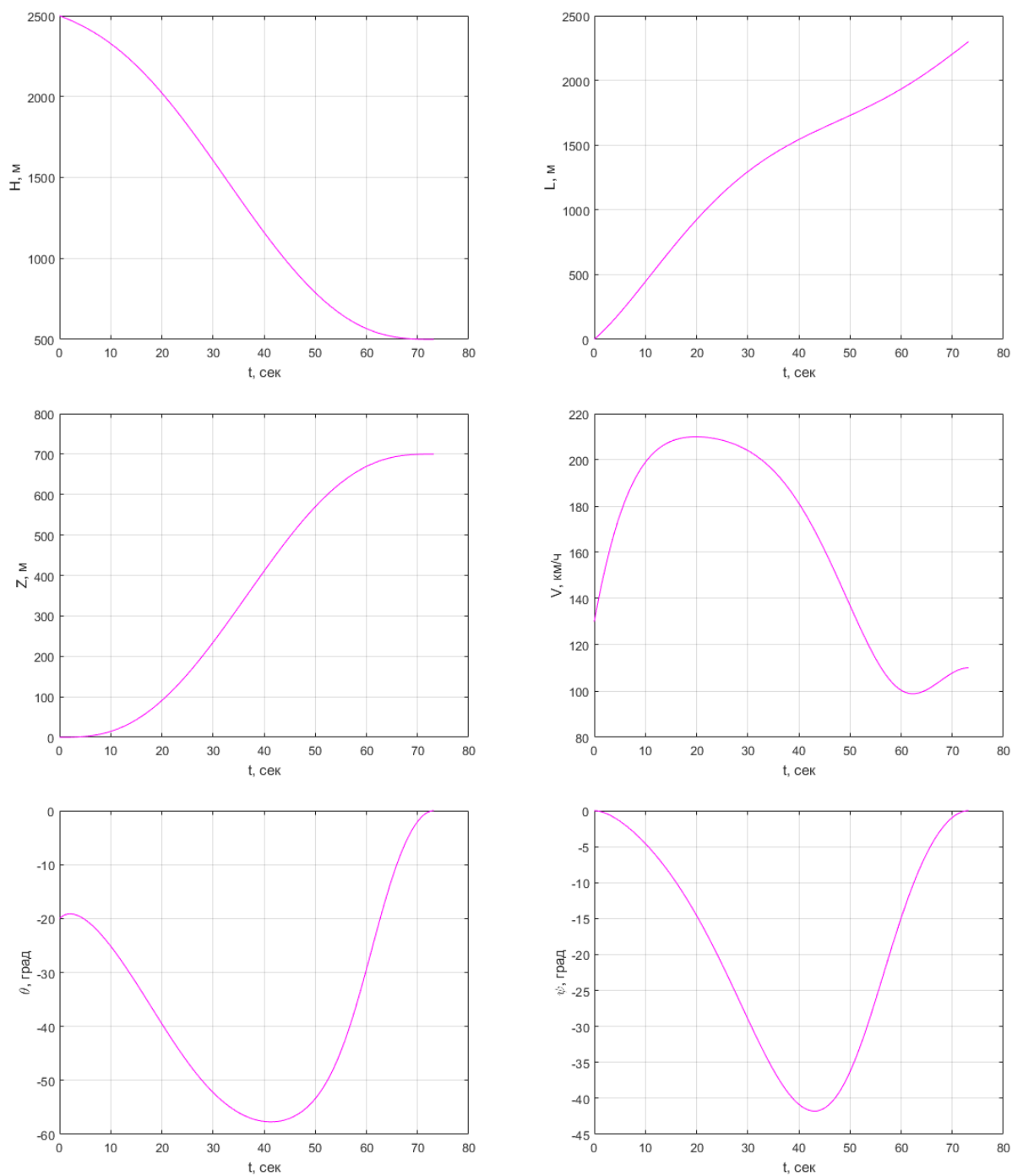


Рис. 4: Графики изменения во времени переменных состояния

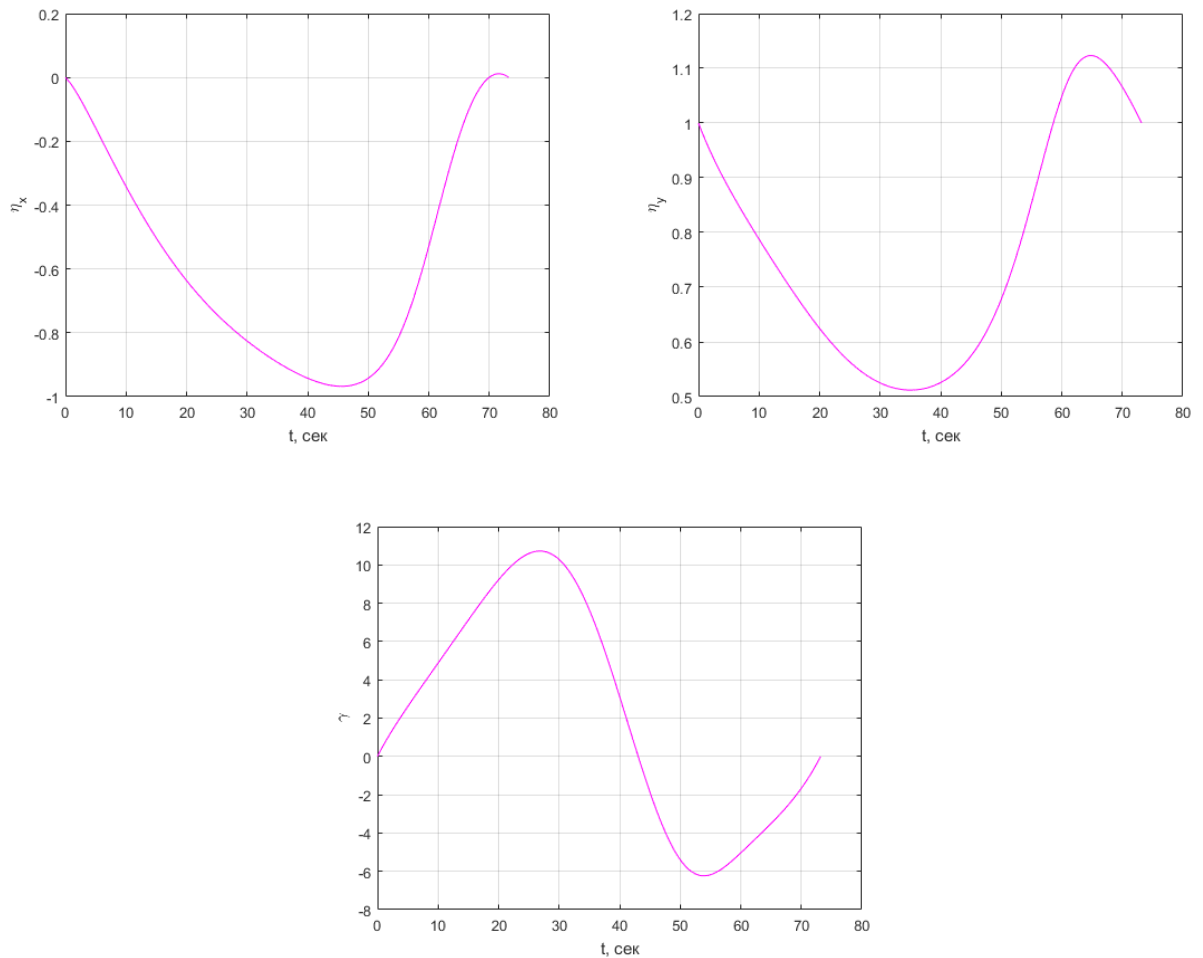


Рис. 5: Графики изменения управлений с течением времени

9.1.2. Смена эшелона (набор высоты ЛА)

	V (км/ч)	H (м)	L (м)	Z (м)	θ (град)	ψ (град)	γ (град)	η_x	η_y
при t_0	120	1050	0	0	-20	0	0	0	1
при t_{opt}	160	1770	1400	500	0	0	0	0	1

Таблица 3: Исходные данные для выполнения маневра

При моделировании оптимальной по времени траектории набора высоты летательным аппаратом, получили $t_{opt} = 38.5937$, при заданной точности $\epsilon = 0.001$.

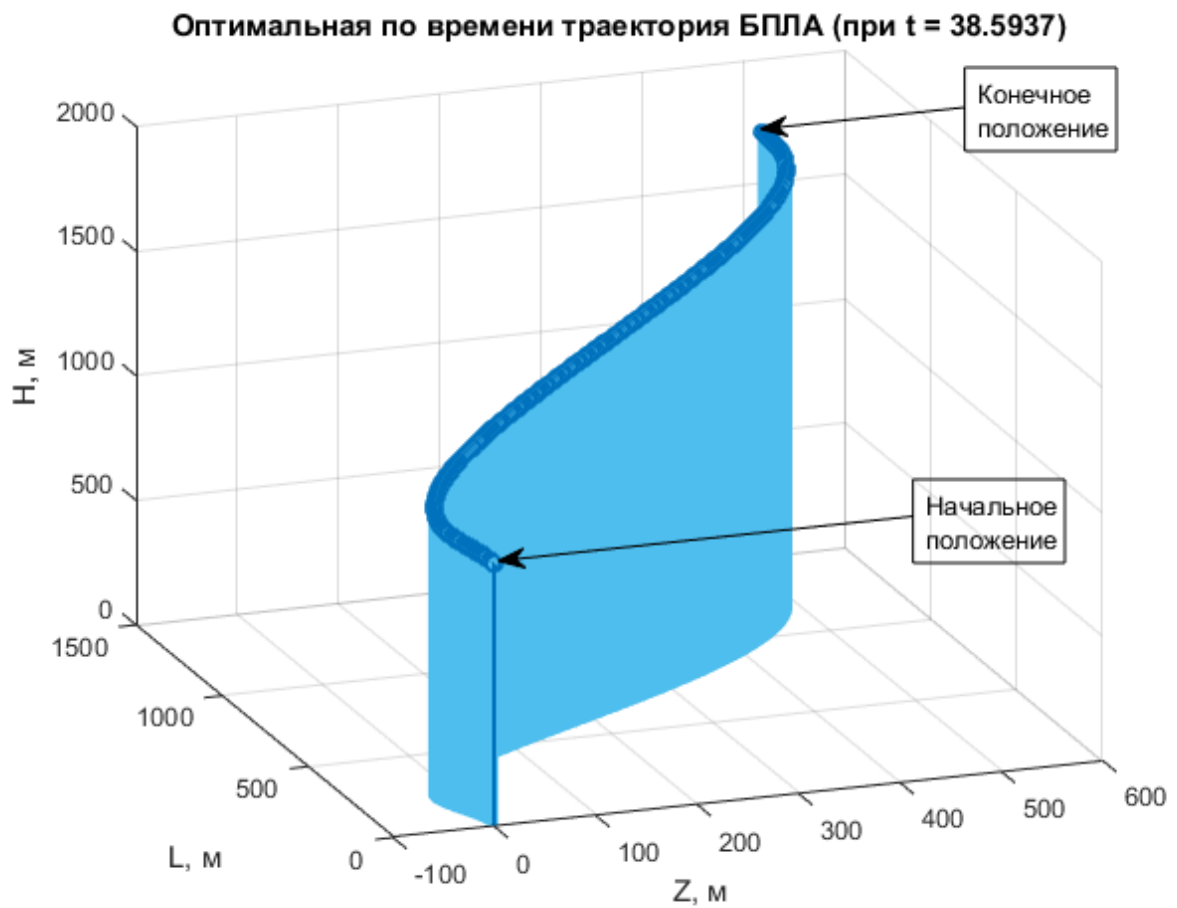


Рис. 6: Оптимальная по времени t траектория движения БПЛА

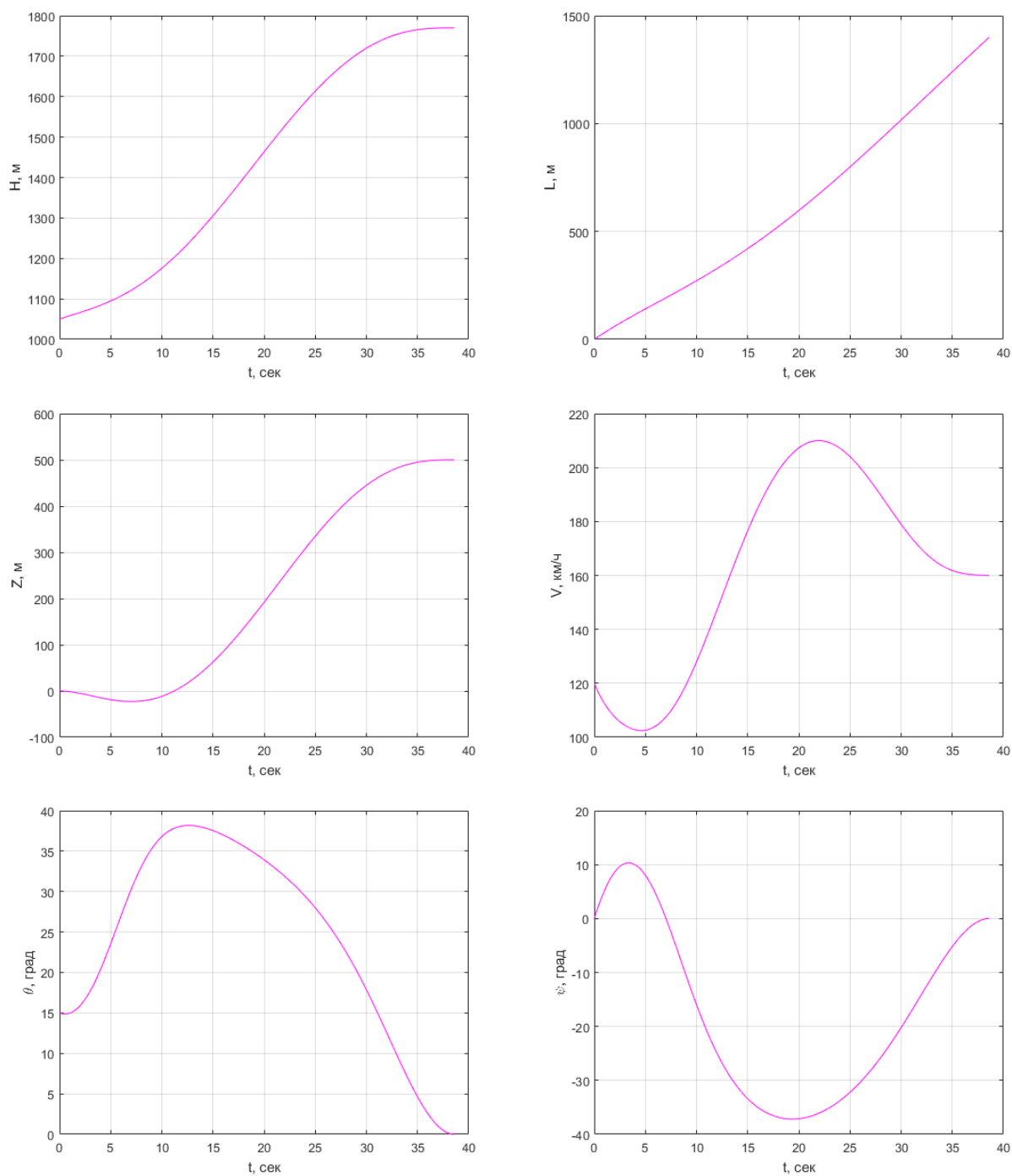


Рис. 7: Графики изменения во времени переменных состояния

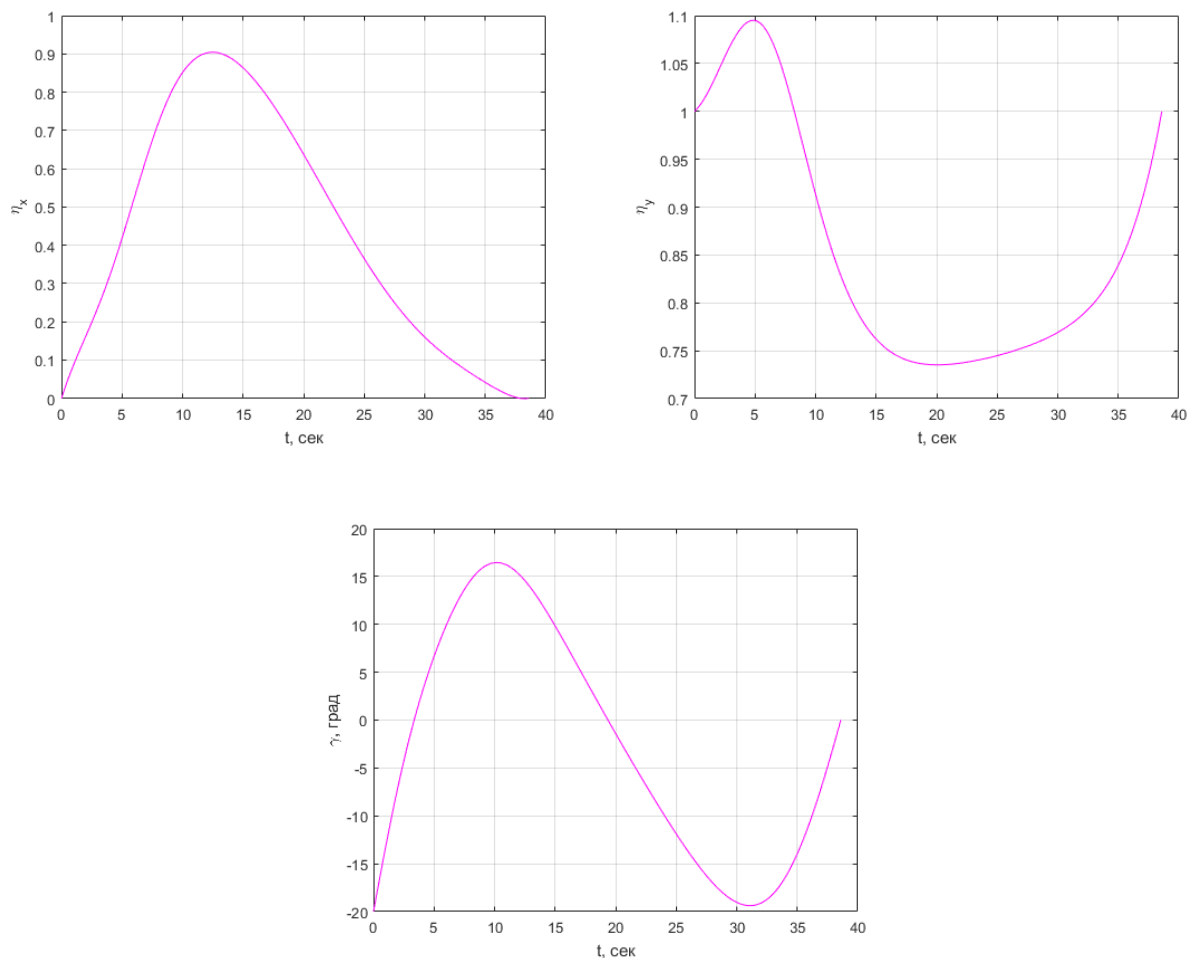


Рис. 8: Графики изменения управлений с течением времени

9.1.3. Разворот БПЛА на 180 градусов

	V (км/ч)	H (м)	L (м)	Z (м)	θ (град)	ψ (град)	γ (град)	η_x	η_y
при t_0	170	850	0	0	0	0	0	0	1
при t_{opt}	170	1150	550	-650	0	180	0	0	1

Таблица 4: Исходные данные для выполнения маневра

При моделировании оптимальной по времени траектории разворота на 180 градусов, получили $t_{opt} = 32.2927$, при заданной точности $\epsilon = 0.001$.

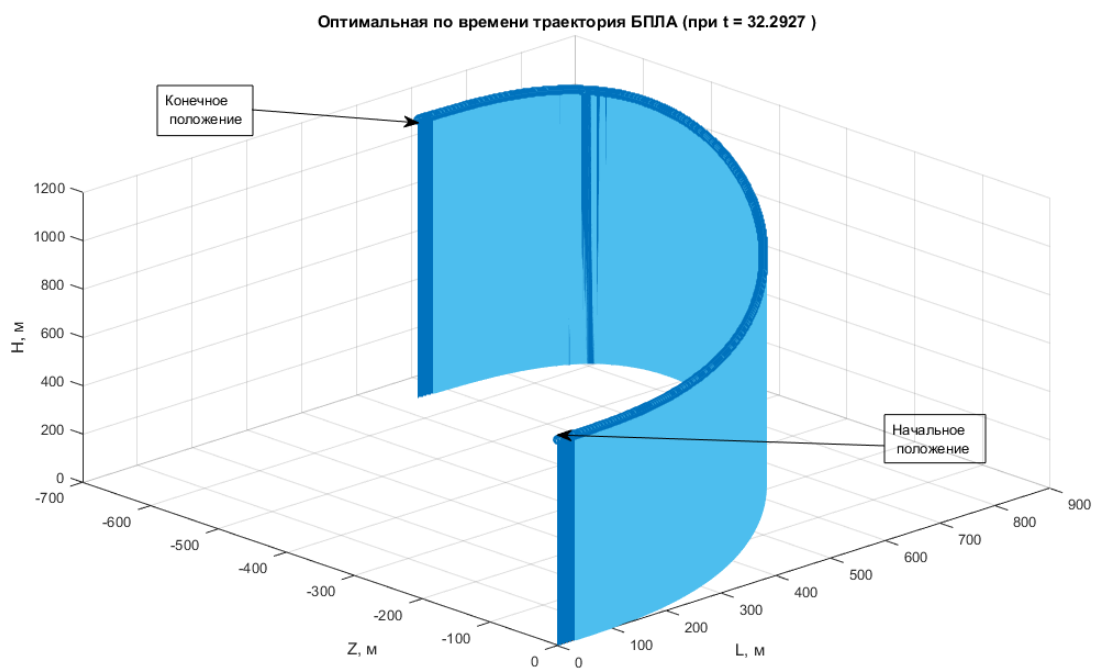


Рис. 9: Оптимальная по времени t траектория движения БПЛА

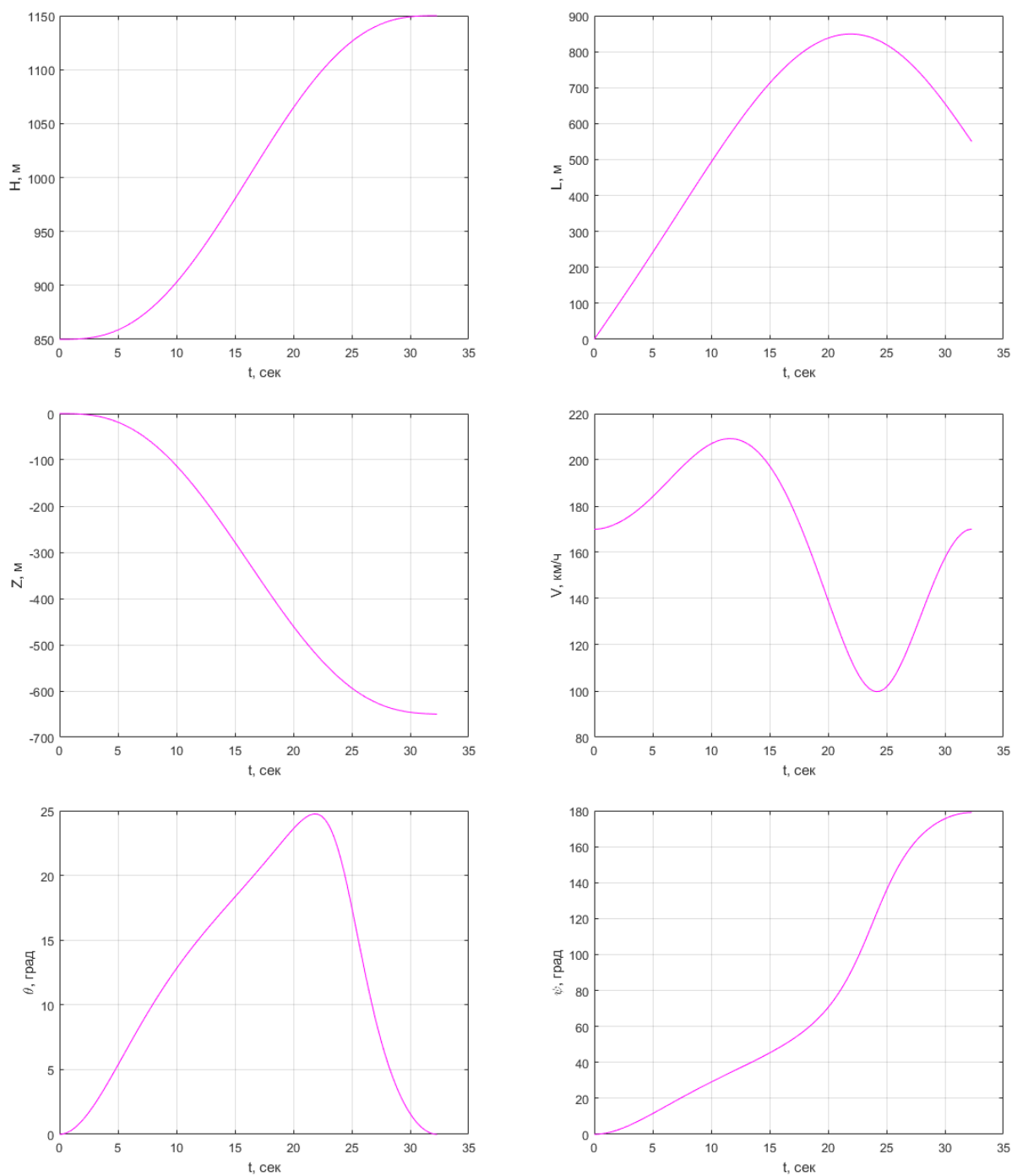


Рис. 10: Графики изменения во времени переменных состояния

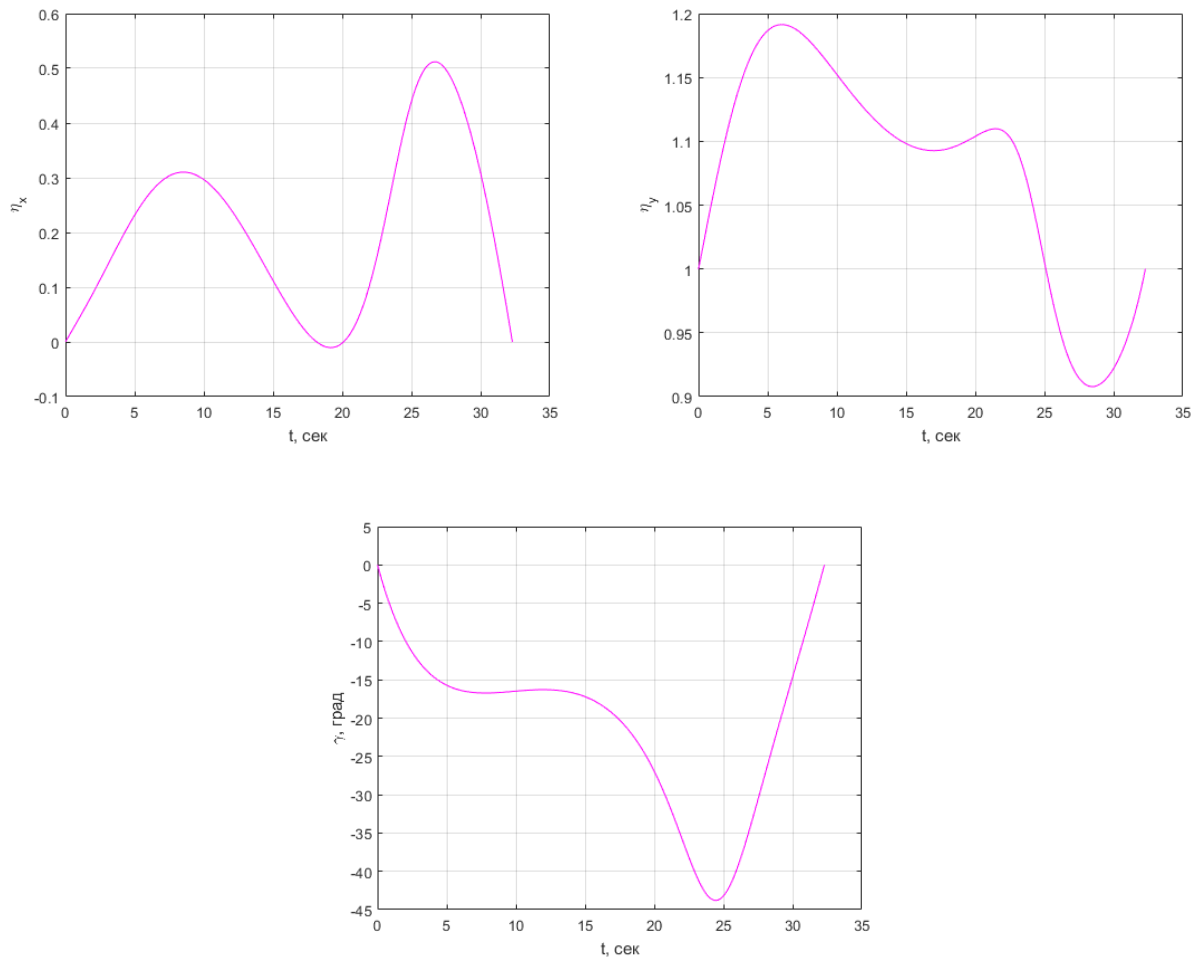


Рис. 11: Графики изменения управлений с течением времени

9.1.4. Смена бокового смещения

	V (км/ч)	H (м)	L (м)	Z (м)	θ (град)	ψ (град)	γ (град)	η_x	η_y
при t_0	155	2550	0	0	0	0	0	0	1
при t_{opt}	155	2551	175	175	0	0	0	0	1

Таблица 5: Исходные данные для выполнения маневра

При моделировании оптимальной по времени траектории смены бокового смещения БПЛА, получили $t_{opt} = 10.2808$, при заданной точности $\epsilon = 0.001$.

Оптимальная по времени траектория БПЛА (при $t = 10.2808$)

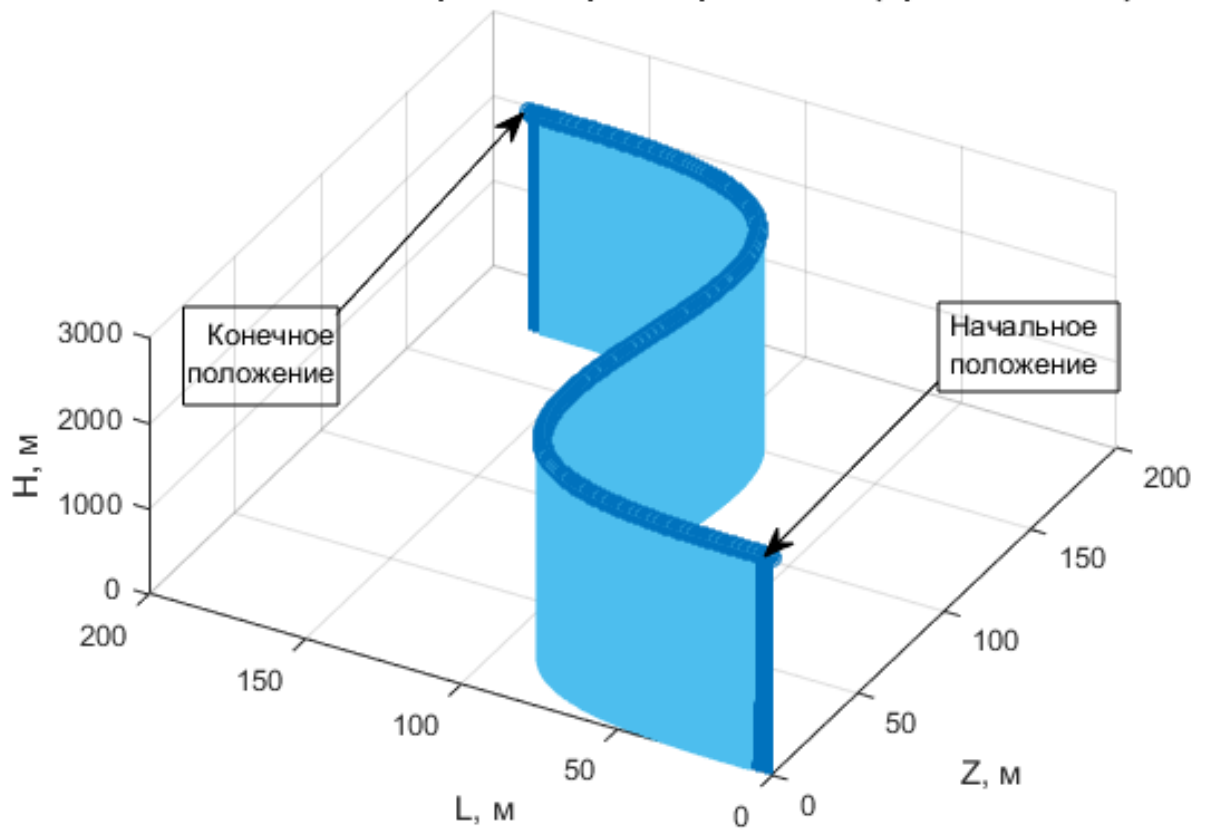


Рис. 12: Оптимальная по времени t траектория движения БПЛА

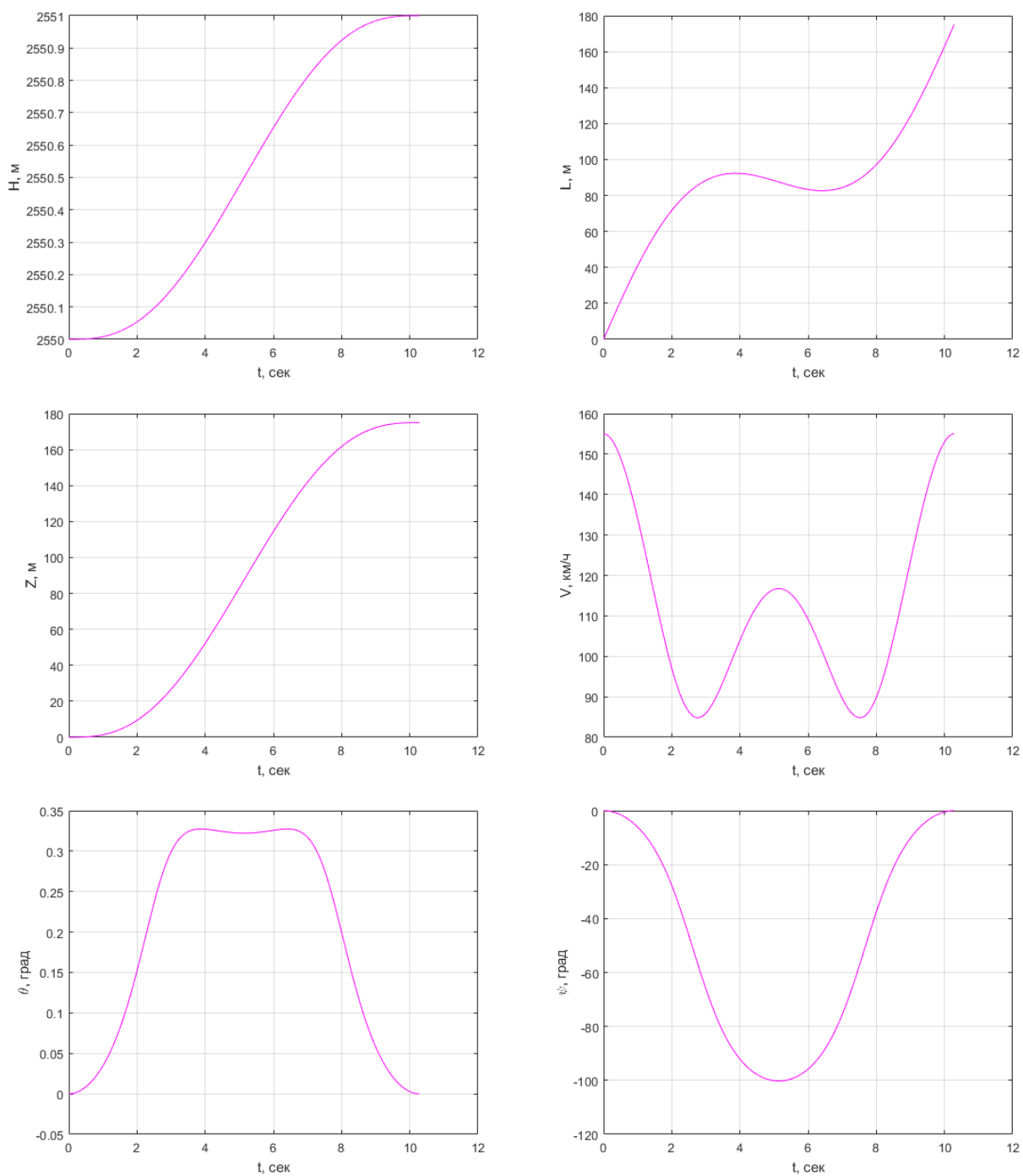


Рис. 13: Графики изменения во времени переменных состояния

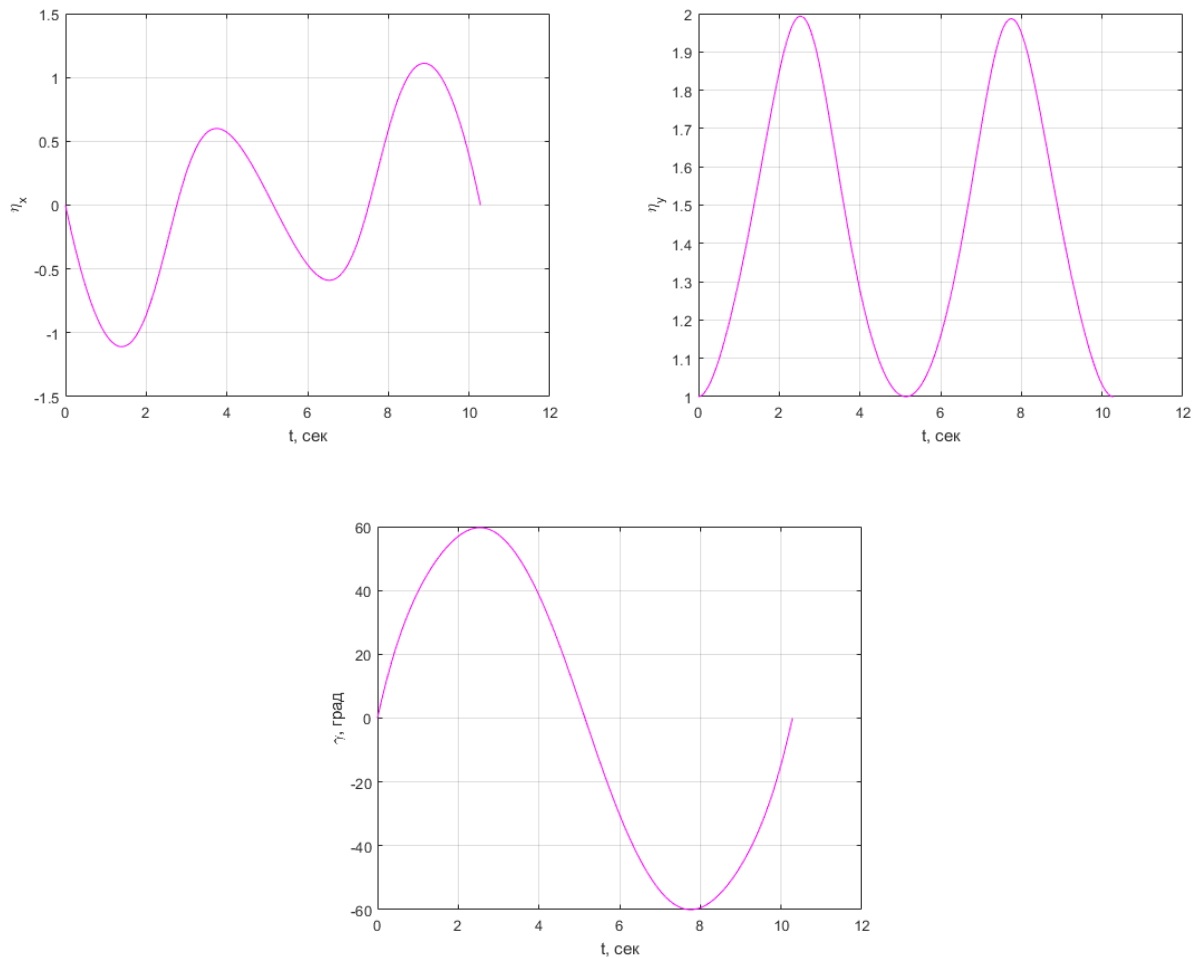


Рис. 14: Графики изменения управлений с течением времени

Выводы

Решение задачи терминального управления при выполнении маневра БПЛА показало, что наибольшее влияние на вид траектории оказывает время полета, что было подтверждено в [2]. Реализуя алгоритм поиска оптимальной по времени траектории, был разработан метод расчета интервала времени при терминальном управлении ЛА.

Разработанный комплекс компьютерного моделирования позволяет наглядно показать поведение БПЛА в процессе маневра, а также изменения фазовых координат и управлений.

При заданной точности $\epsilon = 10^{-3}$ время работы алгоритма в среднем составляло 0.1 сек. Это позволяет говорить о достаточно быстром расчете оптимальной траектории, в случае когда ее вообще реально постро-

ить, учитывая ограничения. Так как все вычисления производились на вполне обычном по производительности домашнем компьютере, можно утверждать о сильном уменьшении времени работы программы, в случае ее переноса на более производительную вычислительную машину. Это также означает практическую применимость в реальных системах автоматического управления БПЛА.

Дополнив созданный программный комплекс, можно обеспечить полностью автоматическое управление летательным аппаратом с возможностью строить сложные траектории, состоящие из базовых маневров.

Заключение

В рамках данной работы особое внимание было уделено:

- рассмотрению литературы по тематике задачи и дальнейшему применению материала в своих вычислениях;
- описанию математической модели управляемого движения БПЛА, основанной на концепции обратных задач динамики, и ее использованию;
- постановке и решению задачи терминального движения летательного аппарата;
- разработке алгоритма поиска оптимального времени полёта для дальнейшего построения пространственной траектории;
- нахождению оптимальной программной траектории и необходимых управлений;
- исследованию управлений в форме обратных связей.

Основным этапом стало создание программного комплекса на основе пакета прикладных задач MATLAB [6], который позволил:

- реализовать разработанный алгоритм поиска оптимальной по времени траектории;
- визуализировать полученные при компьютерном моделировании результаты;
- проанализировать, сравнить поведение БПЛА при различных заданных ограничениях, а также оценить адекватность полученной модели.

Список литературы

- [1] Thomson D.G., Bradley R. Mathematical modelling of helicopter manoeuvres. — Helicopter Soc, 1997.
- [2] Канатников А.Н., Шмагина Е.А. Задача терминального управления движением летательного аппарата. — М. : ФИЗМАТЛИТ, 2010.
- [3] Крищенко А.П. Преобразование многомерных аффинных управляемых систем. — М. : ВНИИСИ, 1991.
- [4] Крутько П.Д. Обратные задачи динамики управляемых систем. Нелинейные модели. — М. : Наука, 1988.
- [5] Крутько П.Д. Обратные задачи динамики в теории автоматического управления. — М. : Машиностроение, 2004.
- [6] Официальный сайт MATLAB. Справочная онлайн документация к вычислительному пакету. — <http://www.mathworks.com/help/>.
- [7] Сайт производителя БПЛА Дозор-600. — 2014. — <http://www.uav-dozor.ru/>.

Приложение

Код функции *trajectory.m*

```
H_0 = 2550;      H_final = 2551;      H_MIN = 250;
                H_MAX = 7500;
L_0 = 0;         L_final = 175;     L_MIN = -25000;
                L_MAX = 25000;
Z_0 = 0;         Z_final = 175;     Z_MIN = -25000;
                Z_MAX = 25000;

V_0 = 155;      V_final = 155;     V_MIN = 70;
                V_MAX = 210;
Nx_0 = 0;       Nx_final = 0;      Nx_MIN = -3.5;
                Nx_MAX = 3.5;
Ny_0 = 1;       Ny_final = 1;      Ny_MIN = 0.5;
                Ny_MAX = 5;

gamma_0 = 0;    gamma_final = 0;   gamma_MIN = -60;
                gamma_MAX = 60;
theta_0 = 0;    theta_final = 0;   theta_MIN = -89;
                theta_MAX = 89;
psi_0 = 0;      psi_final = 0;     psi_MIN = -179;
                psi_MAX = 179;

Time_0 = sqrt((H_final-H_0)^2+(L_final-L_0)^2+(Z_final
              -Z_0)^2)/(V_MAX/3.6);
Time_final=(Time_0+5)*15;
deltaTime=0.5;
eps=0.0001;
flagFound=false;
flagTimeout=false;
optimTime = 0;
curTime=Time_0;
```

```

g=9.80665;
trueVector=true(1,1001);
v_0 = [ Nx_0
        Ny_0*cosd(gamma_0)
        Ny_0*sind(gamma_0) ];
v_Fin=[ Nx_final
        Ny_final*cosd(gamma_final)
        Ny_final*sind(gamma_final) ];
V_0=V_0/3.6;
V_final=V_final/3.6;
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);
y1F=H_final;
y2F=L_final;
y3F=Z_final;
y1Ff=V_final*sind(theta_final);
y2Ff=V_final*cosd(theta_final)*cosd(psi_final);
y3Ff=-1*V_final*cosd(theta_final)*sind(psi_final);
y1Fs=-g+g*v_Fin(1)*sind(theta_final)+g*v_Fin(2)*cosd(theta_final);
y2Fs=g*v_Fin(1)*cosd(theta_final)*cosd(psi_final)-g*v_Fin(2)*sind(theta_final)*cosd(psi_final)+g*v_Fin(3)*sind(psi_final);
y3Fs=-g*v_Fin(1)*cosd(theta_final)*sind(psi_final)+g*v_Fin(2)*sind(theta_final)*sind(psi_final)+g*v_Fin(3)*cosd(psi_final);

```

```

while (( not ( flagFound ) ) && ( curTime <= Time_final ) )
trajectory2 ;
end
if ( curTime > Time_final )
flagTimeout = true ;
end

```

Код функции *trajectory2.m*

```

H_0 = 2550;      H_final = 2551;      H_MIN = 250;
                H_MAX = 7500;
L_0 = 0;        L_final = 175;      L_MIN = -25000;
                L_MAX = 25000;
Z_0 = 0;        Z_final = 175;      Z_MIN = -25000;
                Z_MAX = 25000;

V_0 = 155;      V_final = 155;      V_MIN = 70;
                V_MAX = 210;
Nx_0 = 0;       Nx_final = 0;       Nx_MIN = -3.5;
                Nx_MAX = 3.5;
Ny_0 = 1;       Ny_final = 1;       Ny_MIN = 0.5;
                Ny_MAX = 5;

gamma_0 = 0;    gamma_final = 0;    gamma_MIN = -60;
                gamma_MAX = 60;
theta_0 = 0;    theta_final = 0;    theta_MIN = -89;
                theta_MAX = 89;
psi_0 = 0;     psi_final = 0;     psi_MIN = -179;
                psi_MAX = 179;

Time_0 = sqrt (( H_final - H_0 ) ^ 2 + ( L_final - L_0 ) ^ 2 + ( Z_final
                - Z_0 ) ^ 2 ) / ( V_MAX / 3.6 ) ;
Time_final = ( Time_0 + 5 ) * 15 ;
deltaTime = 0.5 ;

```

```

eps=0.0001;
flagFound=false;
flagTimeout=false;
optimTime = 0;
curTime=Time_0;
g=9.80665;
trueVector=true(1,1001);
v_0 = [ Nx_0
        Ny_0*cosd(gamma_0)
        Ny_0*sind(gamma_0) ];
v_Fin=[ Nx_final
        Ny_final*cosd(gamma_final)
        Ny_final*sind(gamma_final) ];
V_0=V_0/3.6;
V_final=V_final/3.6;
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);
y1F=H_final;
y2F=L_final;
y3F=Z_final;
y1Ff=V_final*sind(theta_final);
y2Ff=V_final*cosd(theta_final)*cosd(psi_final);
y3Ff=-1*V_final*cosd(theta_final)*sind(psi_final);
y1Fs=-g+g*v_Fin(1)*sind(theta_final)+g*v_Fin(2)*cosd(
    theta_final);
y2Fs=g*v_Fin(1)*cosd(theta_final)*cosd(psi_final)-g*

```

```

    v_Fin(2)*sind(theta_final)*cosd(psi_final)+g*v_Fin
    (3)*sind(psi_final);
y3Fs=-g*v_Fin(1)*cosd(theta_final)*sind(psi_final)+g*
    v_Fin(2)*sind(theta_final)*sind(psi_final)+g*v_Fin
    (3)*cosd(psi_final);
while((not(flagFound))&&(curTime<=Time_final))
trajectory2;
end
if(curTime>Time_final)
flagTimeout=true;
end

```