

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

Институт авиационной техники
Кафедра эксплуатации авиационной техники

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«Разработка программы управления горизонтальным полётом гиперзвукового
самолёта для реализации в АБСУ»

По направлению подготовки 25.04.02

«Техническая эксплуатация авиационных электросистем и пилотажно-
навигационных комплексов»
(уровень магистратуры)

Студент _____ Бакри И.

Руководитель ВКР,
_____ Данилин А.И.
степень, звание

Консультант _____

Нормоконтролер _____ Данилин А.И.

Самара 2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

Институт авиационной техники
Кафедра эксплуатации авиационной техники

«УТВЕРЖДАЮ»
Заведующий кафедрой

_____ (Ковалёв М.А.)

«___» _____ 20__ г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ (МАГИСТР)
студенту Бакри Ибрагиму
группа 3222-250402D

Тема работы: «Разработка программы управления горизонтальным полётом гиперзвукового самолёта для реализации в АБСУ»;

утверждена приказом по университету от «___» _____ 20__ г. № ____.

Исходные данные к работе: крейсерское число Маха 4...6; дальность полёта более 7000 км.

Перечень вопросов, подлежащих разработке в работе.

Разработка математической модели движения самолёта.

Формулировка задачи оптимизации траектории.

Определение характеристик изучаемого самолёта.

Анализ результатов расчётов.

Реализация программы управления в АБСУ.

Срок представления законченной работы: «___» _____ 20__ г.

Дата выдачи задания: «___» _____ 20__ г.

Руководитель работы _____ Данилин А.И.

(подпись)

(фамилия, инициалы)

Задание принял к исполнению _____ Бакри И.

(подпись, дата)

РЕФЕРАТ

Пояснительная записка: 99с., 32 рисунка, 3 диаграмма, 7 таблиц, 30 источников, 4 приложения.

ПРОГРАММА УПРАВЛЕНИЯ, ГОРИЗОНТАЛЬНЫЙ КРЕЙСЕРСКИЙ ПОЛЕТ, ГИПЕРЗВУКОВОЙ САМОЛЁТ, РЕАЛИЗАЦИЯ ПРОГРАММЫ УПРАВЛЕНИЯ В АБСУ, ОПТИМИЗАЦИЯ ТРАЕКТОРИИ ПОЛОТА, АЛГОРИТМЫ УПРАВЛЕНИЯ ДВИЖЕНИЕМ ГЛА, УПРАВЛЕНИЯ ПО УГЛУ АТАКИ, СКОЛЬЖЕНИЯ, КРЕНА И ПОЛОЖЕНИЮ ДРОССЕЛЬНОГО КЛАПАНА.

Объектом исследования является управление движения гиперзвукового летательного аппарата на участке горизонтального крейсерского полёта и найти оптимальную траекторию минимизирующая расход топлива между двумя местами.

Цель работы – динамическое проектирование гиперзвукового летательного аппарата, включающее программирование управляемого движения, его оптимизацию и реализация в АБСУ.

В процессе работы использована метод Нелинейного Программирования для оптимизации траектории полёта, и программное обеспечение МАТЛАБ для написания коды управления.

В результате работы определено поведение и значения параметров движения, гарантирующие минимальный расход топлива по программе управления использующей углы атаки, скольжения, крена и положение дроссельного клапана.

Исходными данными является общий пример гиперзвуковой аэродинамической модели (GHAME), этот пример был предоставлен Исследовательским Центром Компьютерного Моделирования НАСА.

Эффективность работы заключается в определении оптимальной траектории полета с минимальным расходом топлива.

ОГЛАВЛЕНИЕ

РЕФЕРАТ.....	3
ОСНОВНЫЕ СОКРАЩЕНИЯ	6
ВВЕДЕНИЕ	7
ГЛАВА 1 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДВИЖЕНИЯ САМОЛЁТА	9
1.1 ГИПЕРЗВУКОВОЙ ЛЕТАТЕЛЬНЫЙ АППАРАТ	9
1.2 ДИНАМИЧЕСКАЯ МОДЕЛЬ САМОЛЁТА	10
1.2.1 Системы координата	10
1.2.2 Уравнения движения	11
1.3 АЭРОДИНАМИЧЕСКИЕ ХАРАКТЕРИСТИКИ ГЛА	16
1.4 ХАРАКТЕРИСТИКИ СИЛОВОЙ УСТАНОВКИ.....	18
1.5 МОДЕЛЬ АТМОСФЕРЫ	19
ГЛАВА 2 ЗАДАЧА ОПТИМИЗАЦИИ ТРАЕКТОРИИ	21
2.1 ФОРМЫ ЗАДАЧИ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ	21
2.2 ПРИНЦИПА МАКСИМУМА ПОНТРЯГИНА.....	22
2.3 ЗАДАЧА ЧИСЛЕННОГО ОПТИМАЛЬНОГО УПРАВЛЕНИЯ	24
2.4 МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ	25
2.5 ПРЯМЫЕ ИЛИ КОСВЕННЫЕ МЕТОДЫ РЕШЕНИЯ ЗОУ	26
2.6 МЕТОД НЕЛИНЕЙНОЙ ОПТИМИЗАЦИИ (НЛП).....	28
2.6.1 Метод решения задачи оптимизации	28
2.6.2 Обычные дифракционные уравнения	32
2.7 ОФОРМЛЕНИЕ ЗАДАЧИ ОПТИМИЗАЦИИ ТРАЕКТОРИИ ПОЛЁТА ГЛА	35
ГЛАВА 3 ХАРАКТЕРИСТИКИ ИЗУЧАЕМОГО САМОЛЁТА	39
3.1 ОПИСАНИЕ ГЛА ПОЛЬЗУЕМЫЙ В ДИССЕРТАЦИИ.....	39
3.2 МАССОВЫЕ СВОЙСТВА	40
3.3 АЭРОДИНАМИЧЕСКИЕ ДАННЫЕ	41
ГЛАВА 4 РЕЗУЛЬТАТЫ И ИХ АНАЛИЗ.....	45
4.1 ПРОГРАММ УПРАВЛЕНИЯ $U = \{A, MA, NT\}$	45
4.1.1 Результаты первой траекторий АБ	45
4.1.2 Результаты второй траекторий ВГ	50
4.2 ПРОГРАММ УПРАВЛЕНИЯ $U = \{A, B, MA, NT\}$	54
4.2.1 Результаты второй траекторий АБ	55
4.2.2 Результаты второй траекторий ВГ	56
ГЛАВА 5 РЕАЛИЗАЦИЯ ПРОГРАММЫ УПРАВЛЕНИЯ В АБСУ	58

5.1 СИСТЕМА АБСУ:	58
5.2 ВХОДЯЩИЕ И ВЫХОДЯЩИЕ ПАРАМЕТРЫ ПРОГРАММЫ УПРАВЛЕНИЯ	58
5.3 РЕАЛИЗАЦИЯ В АБСУ	59
ЗАКЛЮЧЕНИЕ	61
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	63
ПРИЛОЖЕНИЕ А: ГИПЕРЗВУКОВЫЕ СИЛОВЫЕ УСТАНОВКИ	66
ПРИЛОЖЕНИЕ Б: РАСШИРЕННЫЕ УРАВНЕНИЯ	68
ПРИЛОЖЕНИЕ В: КОДЫ МАТЛАБ	69
В.1 ОСНОВНЫЙ КОД	69
В.2 КОД УРАВНЕНИЙ И ОГРАНИЧЕНИЙ	73
В.3 КОД ОБЩЕЙ НАСТРОЙКИ	82
В.3 КОД ФОРМИРУЮЩИЙ ТРАЕКТОРИИ НА МОДЕЛИ ЗЕМЛИ	83
В.4 ДОПОЛНИТЕЛЬНЫЕ СТРОКИ ПРИ НАЛИЧИИ УГЛА СКОЛЬЖЕНИЯ	84
В.5 ДОПОЛНИТЕЛЬНЫЕ КОДЫ	85
ПРИЛОЖЕНИЕ Г: УРАВНЕНИЯ ЗАДАЧИ ПОНТРЯГИНА	86
Г.1 ПРОГРАММА УПРАВЛЕНИЯ УГЛА АТАКА	89
Г.1.1 Оформление функцию Гамильтониана	89
Г.1.2 УРАВНЕНИЯ СОВМЕСТНОГО СОСТОЯНИЯ	91
Г.1.3 УСЛОВИЕ ОПТИМАЛЬНОСТИ	93
Г.2 ПРОГРАММА УПРАВЛЕНИЯ УГЛА СКОЛЬЖЕНИЯ	94
Г.2.1 Оформление функцию Гамильтониана	94
Г.2.2 УРАВНЕНИЯ СОВМЕСТНОГО СОСТОЯНИЯ	95
Г.2.3 УСЛОВИЕ ОПТИМАЛЬНОСТИ	97

ОСНОВНЫЕ СОКРАЩЕНИЯ

ГЛА – гиперзвуковой летательный аппарат,

ЛА – летательный аппарат,

СУ – силовая установка,

НЛП – нелинейное программирование,

ЛП – линейное программирование,

ГПС – гиперзвуковые пассажирские самолёты,

ЗОО – задача оптимального управления,

ГЯБ – Гамильтона-Якоби-Беллмана,

ДТКЗ – двухточечная краевая задача,

ОУ – оптимальное управление,

МВТ – метод внутренней точки,

ККТ – Условия Каруша – Куна – Таккера,

GHAME – общий пример гиперзвуковой аэродинамической модели (Generic Hypersonic Aerodynamic Model Example),

АБСУ – Автоматическая бортовая система управления,

ССП – Системы сервоприводы,

СПУ – Системы штурвального управления,

САУ – Системы автоматического управления,

СТУ – Система траекторного управления,

АТ – Автомата тяги,

СОТ – Система оптимизации траектории

ВВЕДЕНИЕ

Рождение задачи оптимального управления произошло в конце 17 века с проблемой, поставленной Бернулли И., «Какова такая форма проволоки, что шарик, скользящий по ней, проходит расстояние между конечными точками шарика за минимальное время?» Решения были разработаны некоторыми из величайших математических умов в истории, включая братьев Бернулли, Лейбница и Ньютона. Их работа над этой проблемой также породила вариационное исчисление. В более современное время как принцип максимума Понтрягина, так и динамическое программирование Беллмана дали инженерам продвинутые инструменты для более полной разработки оптимальных концепций управления и инструментов [18-20].

Задача оптимального управления – расширение вариационного исчисления и математический метод оптимизации для получения параметров управления, и оптимальное управление можно рассматривать как стратегию управления в теории управления. Решение задачи оптимального управления нелегко, где кусочки задачи разбросаны по многим различным дисциплинам. есть две основные части успешного оптимального управления, первая часть – метод «оптимизации». Вторая часть – метод «дифференциальных уравнений». Сталкиваясь с проблемой оптимального управления, заманчиво просто вставить пакеты оптимизации и численного интегрирования вместе.

В первой главе посмотрели известные системы координаты, и уравнения движения самолёта, которые написали в системе координаты траектории и связанной системе координаты. Модель атмосфера и ускорения силы тяжести написали в качестве изменения плотности воздуха и ускорения силы тяжести с высотой. Написали уравнения аэродинамический сил в качестве функции зависть от угла атака и скольжения, и число маха.

В второй главе посмотрели общие задачи оптимизации и все их формы. Написали подходы и методы решение этой задачи; принцип максимума

Понтрягина, метод нелинейного программирования и метод Гамильтона-Якоби-Беллмана. Написали алгоритм внутренней точки для решения задачи оптимизации в форме НЛП, и метод дискретизации и решения обычных дифференциальных уравнений вместе с задачей оптимизации.

Характеристики изучаемая самолёта поставили в третьей главе. Посмотрели характеристики и величины массы, общих размеров самолёта и моментов инерции. Дата аэродинамических сил и моментов поставили в форме графов зависят от угла атаки и числа Маха.

В четвертой главе поставили результаты двух предлагаемых траектории. Результаты поставили в виде графов и включают в себя: высота, скорость, горизонтальный и вертикальный углы наклона траектории, параметры управления (угол атаки, положение дроссельного клапана и угол крена), углы широты и долготы отдельные и вместе в качестве траектории на карте мире. Первая предлагаемая траектория имеет длина около 8,000 км, а вторая около 15,000 км.

Общее знание о двигателях ГЛА поставили в приложении А. В этом приложении тоже поставили значения удельного импульса. В приложении Б поставили коды МАТЛАБ для задачи минимизации топлива на участке горизонтального крейсерского полёта.

Глава 1 Математическая модель движения Самолёта

Для построения математической модели движения гиперзвукового летательного аппарата (ГЛА), нужно вначале записать уравнения движения ГЛА в общей форме, а затем упростить модель (если нужно) исходя из рассматриваемого режима полёта. Для того требуется определить аэродинамические силы, моменты, действующие на ГЛА, а также силы и моменты от силовой установки.

1.1 Гиперзвуковой летательный аппарат

Гиперзвуковые летательные аппараты - это аппараты, летающие с числом Маха в районе 5 – 25 [2,7]. В этой диссертации будем рассматривать только один тип ГЛА: Гиперзвуковые пассажирские самолёты (ГПС).

Гиперзвуковые пассажирские самолёты предназначены для беспосадочной доставки пассажиров и грузов на дальние расстояния за время, не превышающее 3-х часов, т.е. ГПС сокращают время полёта. При сравнении гиперзвуковых самолётов с сверхзвуковыми и дозвуковыми самолетами по высоте полёта, гиперзвуковые самолёты летают на крейсерских высотах до 40 км, а сверхзвуковые самолеты на высотах 17-24 км и дозвуковые самолеты на высотах 10-11 км, [12]. Для сравнения по скорости и времени полёта рассмотрим следующий пример:

При полёте на дальность 8000 км, дозвуковой самолёт с $M_{\text{крейс}} = 0,8$ затрачивает около 10 часов, сверхзвуковой самолет с $M_{\text{крейс}} = 2$ затрачивает около 4 часов, а гиперзвуковой самолёт с $M_{\text{крейс}} = 6$ затрачивает около полутора часов.

1.2 Динамическая модель самолёта

1.2.1 Системы координата

Требуется несколько систем координаты, приведены на рисунке 1.1, [23]. Преобразование инерциальной системы координаты в систему координаты Земли определяется как T_i^e , где θ – широта к северу от экватора, а Φ – долгота к востоку от произвольной системы отсчета.

$$T_i^e = \begin{bmatrix} -\sin(\phi) & \cos(\phi) & 0 \\ -\sin(\theta)\cos(\phi) & -\sin(\theta)\sin(\phi) & \cos(\theta) \\ \cos(\theta)\cos(\phi) & \cos(\theta)\sin(\phi) & \sin(\theta) \end{bmatrix} \quad (1.1)$$



Рисунок 1.1. системы координаты и преобразование между ними
гунт и вунт – горизонтальный и вертикальный угол наклона траектории

Преобразование из системы координаты Земли в систему координаты скорости определяется как T_e^v , где Ψ – угол направления, а γ – угол подъема.

$$T_e^v = \begin{bmatrix} \cos(\gamma)\cos(\psi) & \cos(\gamma)\sin(\psi) & \sin(\gamma) \\ -\sin(\psi) & \cos(\psi) & 0 \\ -\sin(\gamma)\cos(\psi) & -\sin(\gamma)\sin(\psi) & \cos(\gamma) \end{bmatrix} \quad (1.2)$$

Преобразование из системы координаты скорости в систему координаты устойчивости определяется как T_v^S , где σ – угол крена относительно вектора скорости, а β – угол рыскания,

$$T_v^S = \begin{bmatrix} \cos(\beta) & \cos(\sigma) \sin(\beta) & \sin(\beta) \sin(\sigma) \\ -\sin(\beta) & \cos(\beta) \cos(\sigma) & \cos(\beta) \sin(\sigma) \\ 0 & -\sin(\sigma) & \cos(\sigma) \end{bmatrix} \quad (1.3)$$

Преобразование из системы координаты устойчивости в связанную систему координаты определяется как T_s^b , где α – угол атаки.

$$T_s^b = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \quad (1.4)$$

Поскольку эти вращения между ортогональной системы координат, эти матрицы (1.1-1.4) ортонормированной матрицы преобразования, и обратная трансформация только транспонирование заданной матрицы [3]. Например, $T_b^S = T_s^{b^T}$. Матрицы преобразования также могут быть выражены в нескольких опорных координат для сокращения номенклатуры. Например, преобразование из Земли в связанную систему координаты $T_e^b = T_s^b T_v^S T_e^v$, аналогичным образом, преобразование от связанной системы координаты в систему координаты Землю транспонированная матрица этого преобразования $T_b^e = T_v^e T_s^v T_b^S$.

1.2.2 Уравнения движения

Для гиперзвуковых ЛА необходимо записывать уравнения движения с учётом центробежного ускорения и ускорения Кориолиса от вращения Земли. Поэтому здесь используется полярная система, где существует три типа системы уравнения движения, отличающиеся по выбранным параметрам состояния. Будем писать уравнение поступательного движения в системе координата

траектории и вращательного движения в связанной системе координат, это упростит процедуру численного решения.

Все символы, коэффициенты и системы координат вращательного движения записаны по американской системе, см. Рис. 1.2. На котором показана связанная система координат: ось ox направлена вдоль продольной оси самолёта, ось oy в сторону правой консоли крыла и ось oz вниз. Вокруг оси ox действуют вращательные параметры: угол крена ϕ' , вращательная скорость p , момент крена L , коэффициент момента C_l . В таблице 1 описывание назначения всех параметров относительно осей.

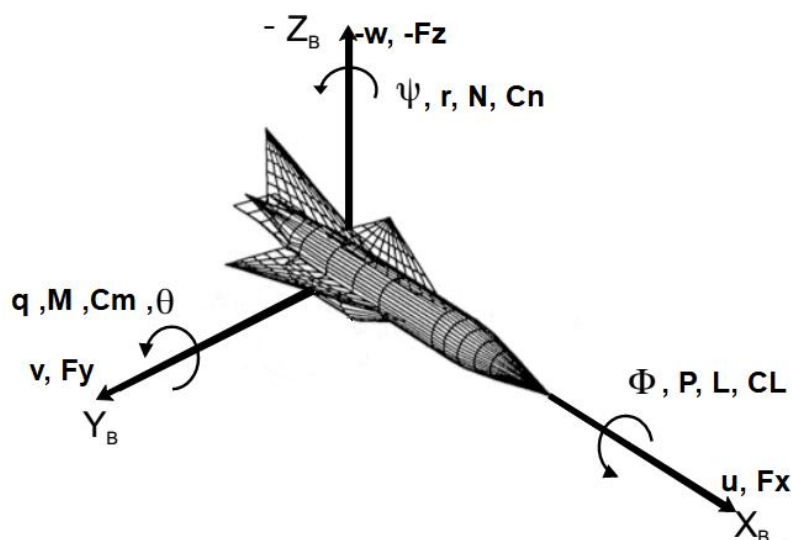


Рисунок 1.2. Американская связанная система координат

Таблица 1.1. Назначения параметров вращательного движения в связанной системе

Параметры движения	Ox	Oy	Oz
Вращательная скорость вокруг	p	q	r
Угол вращения вокруг	ϕ'	θ'	ψ'
Момент вокруг	L	M	N
Коэффициент Моента	C_l	C_m	C_n

Все символы и коэффициенты поступательного движения записаны по системе координата траектории, см. Рис. 1.3. Где: ось oy направлена вдоль продольной оси самолёта, ось oz в сторону левой консоли крыла и ось ox вверх. Вдоль оси ox действуют поступательные параметры: полная скорость V , продольные силы F_y , коэффициенты продольных сил C_x . В таблице 1 описывание назначения всех параметров относительно осей.

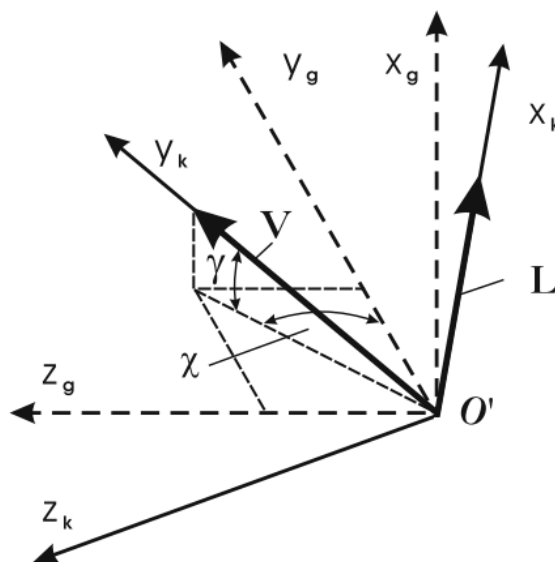


Рисунок 1.3. Траекторная система координат

Таблица 1.2. Назначения параметров поступательного движения в траекторной системе

Параметры движения	OY	OZ	OX
Сила	F_y	F_z	F_x
Коэффициент Силы	C_y	C_z	C_x
Поступательная скорость	V	-	-
Углы системы траектории	μ_a	γ	χ

Математическая модель движения ГЛА определяется, исходя из следующих принятых допущений [5,6]:

- Самолёт симметрично относительно вертикальной плоскости самолёта, т.е. $I_{xy} = I_{yx} = 0$, где x - продольная ось, y - боковая ось и z - вертикальная ось;

- Самолёт рассматривается как твердое тело т.е. $\left(\frac{d\mathbf{r}_b}{dt}\right)_b = 0$, где \mathbf{r}_b - вектор расстояния между любыми двумя точками самолёта;
- Земля вращается вокруг вертикальной оси Z Земли с угловой скоростью $\Omega_e = 2\pi \frac{\text{РАД}}{\text{День}} = 7,27199\text{e} - 5 \frac{\text{РАД}}{\text{с}}$.
- Принято в рассмотрение центробежное ускорение и ускорение Кориолиса от вращения Земли;

Напишем уравнения движения, разделив их на пять групп:

1) Уравнения поступательного движения (уравнения сил):

$$\begin{aligned} \frac{dV}{dt} &= \frac{T_x - D}{m} - g \sin(\gamma) + \omega^2 R \cos^2(\phi) (\sin(\gamma) - \cos(\gamma) \tan(\phi) \sin(\chi)) \\ \frac{d\gamma}{dt} &= \frac{1}{V} \left(\frac{T_z + L \cos(\mu_a) - Y_a \sin(\mu_a)}{m} - g \cos(\gamma) + \frac{V^2}{R} \cos(\gamma) + 2\omega V \cos(\phi) \cos(\chi) \right. \\ &\quad \left. + \omega^2 R \cos^2(\phi) (\cos(\gamma) + \sin(\gamma) \tan(\phi) \sin(\chi)) \right) \\ \frac{d\chi}{dt} &= \frac{1}{V} \left(\frac{T_y + L \sin(\mu_a) + Y_a \cos(\mu_a)}{m \cos(\gamma)} - \frac{V^2}{R} \cos(\gamma) \cos(\chi) \tan(\phi) \right. \\ &\quad \left. + 2\omega V (\tan(\gamma) \cos(\phi) \sin(\chi) - \sin(\phi)) - \frac{\omega^2 R}{\cos(\gamma)} \sin(\phi) \cos(\phi) \cos(\chi) \right) \quad (1.1) \end{aligned}$$

2) уравнения вращательного движения (уравнения моментов):

$$\left. \begin{aligned} I_x \cdot \dot{p}^t - I_{xz} \cdot (p^t q^t + r^t) + q^t r^t \cdot (I_z - I_y) &= L_{\text{аэро}} + L_{\text{тяг}} \\ I_y \cdot \dot{q}^t + r^t p^t \cdot (I_x - I_z) + (p^{t^2} - r^{t^2}) \cdot I_{zx} &= M_{\text{аэро}} + M_{\text{тяг}} \\ I_z \cdot \dot{r}^t + p^t q^t \cdot (I_y - I_x) - I_{xz} \cdot (p^t - q^t r^t) &= N_{\text{аэро}} + N_{\text{тяг}} \end{aligned} \right\} \quad (1.2)$$

3) Уравнения углов Эйлера:

$$\begin{pmatrix} \dot{\phi}' \\ \dot{\theta}' \\ \dot{\psi}' \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi') \tan(\theta') & \cos(\phi') \tan(\theta') \\ 0 & \cos(\phi') & -\sin(\phi') \\ 0 & \sin(\phi') / \cos(\theta') & \cos(\phi') / \cos(\theta') \end{pmatrix} \cdot \begin{pmatrix} p^l \\ q^l \\ r^l \end{pmatrix} \quad (1.3)$$

4) Уравнения траектории Самолёта:

$$\left. \begin{aligned} \dot{h} &= V \sin(\gamma) \\ \dot{\theta} &= V \cos(\gamma) \cos(\chi) / (R \cos(\phi)) \\ \dot{\phi} &= V \cos(\gamma) \sin(\chi) / R \end{aligned} \right\} \quad (1.4)$$

5) Уравнения Расхода Топлива:

$$\dot{m} = -\nu \quad (1.5)$$

Дополнительные уравнения для создания замкнутой группы уравнений 1.1 – 1.5.

$$\begin{pmatrix} p^e \\ q^e \\ r^e \end{pmatrix} = \omega \begin{pmatrix} \cos(\theta') \cos(\psi') \cos(\phi) + \sin(\theta') \sin(\phi) \\ \cos(\phi) (\sin(\phi') \sin(\theta') \cos(\psi') - \cos(\phi') \sin(\psi')) - \sin(\phi') \cos(\theta') \sin(\phi) \\ \cos(\phi) (\cos(\phi') \sin(\theta') \cos(\psi') + \sin(\phi') \sin(\psi')) - \cos(\phi') \cos(\theta') \sin(\phi) \end{pmatrix} \quad (1.6)$$

M_{gb}

$$= \begin{pmatrix} \cos\theta' \cos\psi' & \cos\theta' \sin\psi' & -\sin\theta' \\ \sin\phi' \sin\theta' \cos\psi' - \cos\phi' \sin\psi' & \sin\phi' \sin\theta' \sin\psi' + \cos\phi' \cos\psi' & \sin\phi' \cos\theta' \\ \cos\phi' \sin\theta' \cos\psi' + \sin\phi' \sin\psi' & \sin\theta' \cos\phi' \sin\psi' - \sin\phi' \cos\psi' & \cos\theta' \cos\phi' \end{pmatrix} \quad (1.7)$$

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = M_{gb} \begin{pmatrix} \dot{\theta} \cos(\phi) \\ -\dot{\phi} \\ -\dot{\theta} \sin(\phi) \end{pmatrix} \quad (1.8)$$

$$\begin{pmatrix} p^l \\ q^l \\ r^l \end{pmatrix} = \begin{pmatrix} p^t \\ q^t \\ r^t \end{pmatrix} - \begin{pmatrix} p \\ q \\ r \end{pmatrix} - \begin{pmatrix} p^e \\ q^e \\ r^e \end{pmatrix} \quad (1.9)$$

$$g = g_0 \left(\frac{R_E}{R_E + h} \right)^2, g_0 = 9.80665 \text{ м/с}^2 \quad (1.10)$$

Здесь u, v, w – скорости самолёта относительно системы координат Земли.

p^t, q^t, r^t – полная угловая скорость самолёта относительно системы координат

Земли. p^l, q^l, r^l – угловая скорость самолёта относительно локальной системы

координат. p, q, r – угловые скорости самолёта относительно связанной системы

координат. ϕ', θ', ψ' – углы Эйлера. L, M, N – моменты вокруг осей связанной

системы x, y, z , соответственно. X, Y, Z – аэродинамические силы вдоль осей $x, y,$

z , соответственно. T_y, T_z, T_x – силы тяги вдоль осей y, z, x , соответственно. R – высота полёта самолёта от центра Земли. R_E – радиус Земли. ω – скорость вращения Земли. ϕ – географическая широта. θ – географическая долгота. γ – вертикальный угол наклона траектории. χ – горизонтальный угол наклона траектории. M_{gb} – матрица преобразования от геодезической системы в связанную систему. α, β – углы атаки и скольжения. \dot{v} – расход топлива.

Система уравнений 1.1-1.5 состоит из 13 уравнений с 13 параметров, которые опишут состояние ЛА относительно положения и ориентации траектории. Эти уравнения можно использовать для гиперзвуковых ЛА, космических аппаратов, летающих вокруг Земли или аппаратов, входящие в атмосферу Земли. В этой системе уравнений принято, что Земля имеет эллиптическую форму и система координат связана с этой формой. Она называется геодезической системой координата

Рассмотренные уравнения движения имеют нелинейную связанную форму, поэтому будем использовать программное обеспечение МАТЛАБ для их решения.

1.3 Аэродинамические характеристики ГЛА

Аэродинамика ГЛА моделируются также, как и для самолетов [5]. Существуют три метода для определения аэродинамических характеристик; первый метод через аэродинамические трубы, где требуются три трубы (дозвуковая, сверхзвуковая и гиперзвуковая) для покрытия режимов полета. Сложность пользования результатов продувок в аэродинамических трубах заключается в сопоставлении числа Рейнольдса в аэродинамической трубе с числом Рейнольдса в свободном полете при числе Маха выше 10.

Второй метод определения аэродинамических характеристик проводится через использование вычислительной гидродинамики, но это требует очень дорогих суперкомпьютеров.

Третий метод составляют полуэмпирические методы, которые используют аналитические уравнения и интерполированные базы данных аэродинамических продувок.

Примером последнего метода является панельный метод (Panel Methods), который особенно часто применяется в сверхзвуковых и гиперзвуковых режимах. Для обычных самолётов и ракет пользуются программой (DATCOM), а для сверхзвуковых и гиперзвуковых самолётов пользуются американской программой расчёта обтекания сверхзвукового / гиперзвукового произвольного тела (S/HABP) или методом (PAN-AIR) от НАСА.

Силы обычно рассчитываются по формуле $0,5 \rho V^2 S_{\text{этал}}$ (например: $X = 0,5 \rho V^2 S_{\text{этал}} C_X$), а моменты по формуле $0,5 \rho V^2 S_{\text{этал}} L_{\text{этал}}$ (например: $M = 0,5 \rho V^2 S_{\text{этал}} L_{\text{этал}} C_m$), где ρ обозначает плотность потока, V скорость набегающего потока, $S_{\text{этал}}$ площадь эталонного участка и $L_{\text{этал}}$ эталонная длина.

В скоростной системе координат, коэффициенты подъемной силы, сопротивления и боковой силы обозначены C_L, C_D, C_Y соответственно. В связанной системе координат эти же коэффициенты обозначены через C_Z, C_X, C_Y , а коэффициенты момента крена, момента тангажа, момента рыскания через C_l, C_m, C_n соответственно.

Аэродинамические параметры даны в табличной форме [1], поэтому их необходимо аппроксимировать полиномами второй степени для угла атаки и второй степени для числа Маха.

$$\left. \begin{aligned} C_{xt}(\alpha, M) &= C_{xt0} + C_{xt1}\alpha + C_{xt2}\alpha^2 + C_{xt3}M + C_{xt4}M^2 \\ C_{yt}(\beta, M) &= C_{yt1}\beta + C_{yt2}\beta^2 + C_{yt3}M + C_{yt4}M^2 \\ C_{zt}(\alpha, M) &= C_{zt0} + C_{zt1}\alpha + C_{zt2}\alpha^2 + C_{zt3}M + C_{zt4}M^2 \end{aligned} \right\} (1.11)$$

$$\left. \begin{aligned} C_{lt}(\beta, M) &= C_{lt1}\beta + C_{lt2}\beta^2 + C_{lt3}M + C_{lt4}M^2 + C_{xp}p^* \\ C_{mt}(\alpha, M) &= C_{mt1}\alpha + C_{mt2}\alpha^2 + C_{mt3}M + C_{mt4}M^2 + C_{xq}q^* \\ C_{nt}(\beta, M) &= C_{nt1}\beta + C_{nt2}\beta^2 + C_{nt3}M + C_{nt4}M^2 + C_{xr}r^* \end{aligned} \right\} (1.12)$$

В уравнениях 1.11 и 1.12 видно что коэффициенты сил и моментов C_X, C_Y, C_Z и C_l, C_m, C_n не зависят от отклонений поверхности управления $\delta_e, \delta_r, \delta_a$, от угловых скоростей p, q, r , а также от $\dot{\alpha}, \dot{\beta}$, из-за отсутствия информации о них.

Аэродинамические силы обычно задаётся в системе координата скорости, поэтому требуется преобразование в связанную систему координата, для того пользуются следующее преобразование:

$$\begin{pmatrix} C_X \\ C_Y \\ C_Z \end{pmatrix} = \begin{pmatrix} \cos(\alpha) * \cos(\beta) & -\cos(\alpha)\sin(\beta) & -\sin(\alpha) \\ \sin(\beta) & \cos(\beta) & 0 \\ \sin(\alpha)\cos(\beta) & -\sin(\alpha)\sin(\beta) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} -C_D \\ C_Y \\ -C_L \end{pmatrix} \quad (1.13)$$

Аэродинамические характеристики и характеристики изучаемое самолёта предложены в главе 4.

1.4 Характеристики Силовой Установки

Силовая установка — это средства удержания ракет и самолетов в воздухе, где вектор тяги преодолевает сопротивление и силу тяжести и поддерживает скорость необходимую для подъема. Вектор тяги обычно направлен параллельно оси аппарата, или с фиксированным углом. При применении второго закона Ньютона к потоку выхлопных газов от двигателя со скоростью C и массовым расходом \dot{v} получим [5]:

$$v = \frac{T}{I_{\text{уи}}(M)g} \quad (1.14)$$

Где T — сила тяги, $I_{\text{уи}}$ — удельный импульс, v — расход топлива, g — ускорение силы тяжести, скорость потока выхлопных газов $c = I_{\text{уи}}g$. Здесь удельный импульс изменяется относительно числу Маха и найдем в таблицах, приложение А. $v_{\text{ном}}$ — Номинальный расход топлива. $v_{\text{макс}}$ — Максимальный расход топлива. η_t — Положение дроссельного клапана [0-1]. $v_{\text{ном}}$ и $v_{\text{макс}}$ заданные в приложении А,

а η_t определяем при решении задачу оптимального управления, в качестве параметр управления ($u = \eta_t$).

Чтобы считать расход топлива нужно знать силу тяги, поэтому сила тяги моделируется по следующему уравнению [11]:

$$T = (v_{\text{ном}} + (v_{\text{макс}} - v_{\text{ном}}) * \eta) I_{\text{уи}}(M) g \quad (1.15)$$

Здесь: удильный импульс найдем в таблицах, см. приложение А, рис А.2.

1.5 Модель атмосферы

Характеристики атмосферы, такие как плотность, температура и давление, влияют на траекторию аэрокосмического аппарата, где плотность воздуха определяет аэродинамические силы и моменты, температура связана со скоростью звука, а давление воздуха модулирует тягу ракетного двигателя, Температура воздуха меняется с высотой, но уменьшается не пропорционально с увеличением высоты.

Атмосфера делится на несколько слоев [5]. Тропосфера от уровня моря до 11 км, тропопауза до 20 км, стратосфера до 80 км; выше плотность воздуха настолько низка что она не может обеспечить никакой значительной аэродинамической силы.

В стандартной атмосфере с высотой от 0 до 40 км, плотность атмосферы вычисляется по следующему закону [9]:

$$\left. \begin{aligned} \rho &= \rho_0 \exp\left(-\frac{h}{H(h)}\right) \\ H(h) &= 10351.8 - (0.0368512)h - (1.02368e - 5)h^2 + (2.63363e - 10)h^3 \end{aligned} \right\} \quad (1.16)$$

Где h – высота самолёта, ρ_0 – плотность воздуха на поверхности Земли. ρ - плотность воздуха на высоте h .

Для расчетов, ограничивающие в нижних слоях атмосферы, используемых простых функций, например, международного стандарта атмосфера-1962, или ISO-2533 [5]:

$$\left. \begin{aligned} T &= 288.15 - 0.0065 h \\ P &= 101325 \left(\frac{T}{288.15} \right)^{5.2559} \end{aligned} \right\} \text{где } h < 11 \text{ km} \quad (1.17)$$

$$\left. \begin{aligned} T &= 216 \\ P &= 22630 e^{(-0.00015769(h-11000))} \end{aligned} \right\} \text{где } 11 \text{ km} < h < 80 \text{ km} \quad (1.18)$$

$$\rho = \frac{P}{R T} \quad (1.19)$$

Где T – температура на высоте h , P – давление на высоте h , R - универсальная газовая постоянная.

Для сравнения изменения плотности обеих моделей с высотой, эти модели показаны на Рисунке 3.

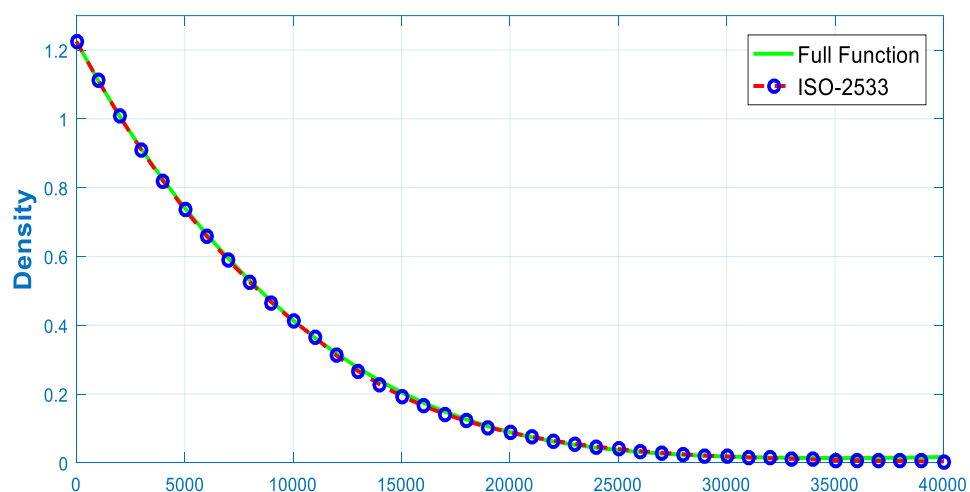


Рисунок 1.4. Сравнение функций плотности атмосфера

Из Рисунка 1.4, видно, что обе модели близки друг к другу и можно брать любую; исходя из этого берём первую модель (Full-Function) - уравнение (1.16), из-за его простоты и точности.

Глава 2 Задача оптимизации траектории

Задача оптимизации траектории — это специфическое применение теории оптимального управления, и формулировка задачи оптимизации траектории одинакова для общей задачи оптимального управления.

Задачу оптимизации это искать программу управления u , который создать воздействия на систему движения, и минимизирует расход топлива v или время полёта, где минимизация \dot{v} достигается при максимизации функцию эффективности « J », т.е. при: $J = -v_f \Rightarrow \text{Max}(J) = -\text{Min}(v_f)$.

В этой диссертации взяли $u = \{\alpha, \mu_a, \eta_t\}$ для программы управления движения, и при наличии время посмотрим влиянию $u = \{\alpha, \beta, \mu_a, \eta_t\}$ на результатов, в качестве программы управления движения. Где α – угол атака, β – угол скольжения, μ_a – угол крена в системе координата траектории, η_t – положение дроссельного клапана [0-1].

2.1 Формы задачи оптимального управления

Задача оптимального управления можно писать в трех разных форм [18-22]: форма Больца, форма Лагранжа и форма Майера. Форма Больца имеет два термина первый называется термином Майера «М», а второй называется Лагранжа « l ».

Минимизация:

$$J(u) = M(x(T), T) + \int_0^T l(x, u, t) dt \quad (2.1)$$

подлежащий к:

$$\left. \begin{aligned} \dot{x}(t) &= f(x(t), u(t), t) \\ x(t_0) &= x_0 \\ f(x_f, t_f) &= 0 \\ p_{\max} &\geq x_f \geq p_{\min} \end{aligned} \right\} \quad (2.2)$$

$\dot{x}(t)$ – динамическая система, $x(t_0)$ – начальное условия, $f(x_f, t_f)$ – конечное условия, T – конечное время. J здесь называется функциональным или функциональным эффективности, и может быть, например, масса топлива, ползавшая для пуска спутника к орбите, или время поездки между двумя точкой.

Форма Лагранжа имеет единый термин (термин Лагранжа):

Минимизация:

$$J(u) = \int_0^T l(x, u, t) dt \quad (2.3)$$

подлежащий к уравнениям и условиям Больца (2.2). уравнение (2.3) называется функцией эффективности траектории, и l здесь может быть например, расстояние между двумя точкой.

Форма Майера также имеет единый термин (термин Майера):

Минимизация:

$$J(u) = M(x(T), T) \quad (2.4)$$

подлежащий к уравнениям и условиям Больца (2.2). уравнение (2.4) называется функцией конечной эффективности, где M может быть время пуска спутника к орбите, и задача оптимального управления здесь это минимизировать этой время.

2.2 Принципа максимума Понтрягина

Функция оптимизации траектории можно формировать на базе принципа максимума Понтрягина [10,13,14], который заключается в максимизации значение функции эффективности « J » под данными конечными P_f условиями.

$$J = \int_{t_0}^{t_f} g(x, u, t) dt \quad (2.5)$$

$$P(x_f, u_f, t_f) = 0 \quad (2.6)$$

Функция Гамильтона $H = \{\psi, f(x, u, t)\}$ этой задачи имеет следующую форму:

$$H(\psi, x, u, t) = g + \sum_{i=1}^n \psi_i f_i(x, u, t) \quad (2.7)$$

Где ψ_i - сопряженные переменные (множители Лагранжа), n – число функций состояния. Согласно принципу максимума Понтрягина, уравнения Эйлера для задачи оптимального управления могут быть записаны с использованием функции Гамильтона следующим образом:

$$\dot{\psi} = \frac{-\partial H}{\partial x} \quad (2.8)$$

$$\dot{x} = \frac{\partial H}{\partial \psi} \quad (2.9)$$

Оптимальное управление u для каждого t должно составлять максимум Гамильтониана (2.10), где функция управления удовлетворяет условие (2.11).

$$M(\psi, x, t) = \max_{u \in U} H(\psi, x, u, t) \quad (2.10)$$

$$\frac{\partial H}{\partial u} = 0 \text{ и } \frac{\partial^2 H}{\partial u^2} < 0 \quad (2.11)$$

Здесь U – область управления.

Оптимизации с помощью принципа максимума Понтрягина является сложной задачей и требуется длинные символьные расчёты. В этой диссертации начали решения задачу оптимизации с помощью принцип Понтрягина, но уравнения задачи стали многими и длинными, и для их решения построили решатель в МАТЛАБ на базе метода конечных-разностей, но решения не сходятся, и требует добивать длинные коды для регулирования условиях Понтрягина, поэтому решили брать другую подход решения задачи оптимизации и было метод нелинейное программирование. Все уравнения задачи Понтрягина поставили в приложении Г.

2.3 Задача численного оптимального управления

Задача численного оптимального управления (ЗОУ) – это задача ограниченной оптимизации с динамической системой в качестве ограничения.

Динамическая система имеет следующую форму, [18-21]:

$$\dot{x}(t) = f(x(t), u(t), t), \quad t \in (0, T) \quad (2.12)$$

Здесь: $x(t)$ – параметры состояния динамической системы. $u(t)$ – параметры управления.

Задача ЗОУ заключается в искать параметрам $u(t)$, минимизирующее целевую функцию «J», (2.13), подлежащий динамическую систему (2.12).

$$J(u) = M(x(T)) + \int_0^T l(x, u, t) dt \quad (2.13)$$

M – термин Майера. l – термин Лагранжа.

Ограниченные динамической системы может иметь следующие формы (a), (b) или (c) в (2.14).

$$g(x(t), u(t), t) \geq 0 \text{ (a)}, \quad h(x(t), u(t), t) = 0 \text{ (b)}, \quad \int_0^T e(x(t), u(t), t) dt = 0 \text{ (c)} \quad (2.14)$$

Где интегральные ограничения могут быть устранены путем добавления параметры состояний (2.15). неравенства перепишем в форму (2.16):

$$\int_0^T e_k(x(t), u(t)) dt = 0 \Rightarrow \dot{z}(t) = e_k(x(t), u(t)) \text{ и } z(0) = z(T) = 0 \quad (2.14)$$

$$g_k(x(t), u(t)) \geq 0 \Rightarrow g_k(x(t), u(t)) - \varepsilon_k(t) = 0 \text{ и } \varepsilon_k(t) \geq 0 \quad (2.15)$$

При незнании некоторых постоянных $p = \{p_1, p_2, \dots, p_n\}$ можно поставить их в качестве часть задачи, и динамическая система будет имеет следующую форму (2.16-17):

$$\dot{x}(t) = f(x(t), u(t), p, t) \quad (2.16)$$

$$\dot{p}(t) = 0 \quad (2.17)$$

2.4 Методы решения задачи оптимального управления

Сложность решения задачи ОУ приходит от преопределенных конечных условий (в нашей задаче: $h_f = 30000 \text{ km}, \phi_f, \theta_f$) и здесь задача называется двухточечной-краевой-задачей (ДТКЗ), этот тип задач требует специальные методы для их решения, см. диаграмм 2.1. При отсутствии конечных условий, задача решается сразу при использовании обычных численных методов интеграции (например, методы Эйлера, метод Рунга-Кутты и все другие методы интеграции, решающие задачу начальной-величины)

Для решения задачи оптимального управления есть три подхода, [18-21]: динамическое программирование, не прямые методы и прямые методы. Динамическое программирование построено на базе уравнения Гамильтона-Якоби-Беллмана (ГЯБ). Непрямые методы построены на базе принципа максимума Понтрягина и задача решается в качестве двухточечная-краевая-задача (ДТКЗ). Прямые методы решаются при преобразовании задачи в задачу нелинейного программирования (НЛП). Методы решения этих трех подходов разные и отличаются по сложности. Классификация подходов и методов решения задачи ОУ объяснено в диаграмме 2.1.

По методам решения задачи оптимального управления численно, известные подходы эти прямые методы и динамическое программирование, где не требуют писать длинные совместные дополнительные уравнения как в принципе максимума Понтрягина. Самые известные методы решения задачи эти нелинейные программирования (НЛП) и метод Коллокации.

Напомним, что принцип максимума Понтрягина дает точные результаты, а все другие подходы дают приближительные результаты, но одновременно являются простым методом решения задачи ОУ.

Диаграмма 1.1. классификация подходов и методов решения задачу ОУ



2.5 Прямые или косвенные методы решения ЗОУ

Прямой метод строит последовательность точек x_1, x_2, \dots, x^* такой, что целевая функция минимизирована и обычно $F(x_1) > F(x_2) > \dots > F(x^*)$. Непрямой (косвенный) метод пытается найти корень необходимого условия $F'(x) = 0$. По крайней мере, в принципе прямой метод требуется сравнение значений целевой функции. Напротив, косвенный метод должен вычислить уклон $F'(x)$, а затем решить, достаточно ли он близок к нулю. По сути, косвенный метод пытается

найти корень необходимых условий. Прямой метод пытается найти минимум целевой функции (или функции Лагранжа).

Так почему бы не использовать косвенный метод? [18], на практике существует по меньшей мере две основных трудности, которые отвлекают внимание от косвенного метода:

1. Пользователь должен вычислить количество Гамильтона H_x , H_u . К сожалению, эта операция требует умного пользователя с хотя бы некоторыми знаниями теории оптимального управления. Кроме того, даже если пользователь знаком с необходимой теоретической базой, может быть очень трудно построить эти выражения для сложных приложений. Наконец, подход не является гибким, так как каждый раз, когда ставится новая задача, требуется новый расчет соответствующих производных!
2. Метод сам не является надежным. Одна из трудностей заключается в том, что пользователь должен угадать значения для сопряженных переменных « λ », которые, поскольку они не являются физическими величинами, очень неинтуитивны! Даже при разумном предположении для сопряженных переменных численное решение сопряженных уравнений может быть очень плохо обусловлено! Чувствительность косвенного метода была признана в течение некоторого времени. Вычислительный опыт работы с техникой в конце 1960-х годов обобщен Брайсоном и Хо [54, р. 214]:

Основная трудность этих методов заключается в том, чтобы начать работу; т. е. найти первую оценку неопределенных условий на одном конце, которая дает решение, достаточно близкое к указанным условиям на другом конце. Причина этой специфической трудности заключается в том, что экстремальные решения часто очень чувствительны к небольшим изменениям неопределенных граничных условий.... Поскольку уравнения системы и уравнения Эйлера-Лагранжа связаны вместе, для численного интегрирования с плохо угаданными начальными условиями нет ничего необычного в создании "чужой" траекторий в пространстве состояний. Эти траектории могут

быть настолько чужими, что значения $x(t)$ и/или $\lambda(t)$ превосходят числовой диапазон компьютера!

2.6 Метод нелинейной оптимизации (НЛП)

Нелинейная оптимизация или нелинейное программирование (НЛП) стало необходимым инструментом в инженерии. Метод НЛП является однем из численных методов для решения задачи оптимизации. Задача НЛП требует нахождения конечного числа переменных, таких, что целевая функция «J» без нарушения набора данных ограничений. Задача НЛП часто называются оптимизацией параметров. Важные частные случаи проблемы НЛП включают линейное программирование (ЛП), квадратичное программирование (КП) и задачи наименьших квадратов (НК).

2.6.1 Метод решения задачи оптимизации

Чтобы решить задачу типа НЛП необходимо брать логарифм, решающий таких задач, самый эффективный и известный логарифм, который будем использовать являются методом внутренней точки (МВТ) или барьерным методом. Идея метода внутренних точек - итерационно подойти к оптимальному решению из внутренней части допустимого множества данных функций и условий.

Логарифм внутренней точки (МВТ) имеет следующие этапы, где общая задача оптимизации в методе НЛП, [18,25,26,27]:

$$\left. \begin{array}{l} \text{Минимизация } J(x_i) \\ d_i \geq g_i(x_i) \geq a_i \\ h_i(x_i) = b_i, \quad i = 1 \dots m \end{array} \right\} \quad (2.18)$$

1. Первый этап в логарифме МВТ – это преобразовать не-равенственные условие $(d_i \geq g_i(x_i) \geq a_i)$ в равенственные и добывать переменные, сохраняющие значения данного условия. Все условия должны сначала имеют

одна ограниченная сторона, и неравенство должно быть имеет положительный знак « \geq », потом преобразовать их в условия типа равенства.

$$\begin{pmatrix} g_i(x_i) \geq a_i \\ g_i(x_i) \leq d_i \end{pmatrix} \Rightarrow \begin{pmatrix} g_i(x_i) - a_i \geq 0 \\ g_i(x_i) - d_i \leq 0 \end{pmatrix} \Rightarrow \begin{pmatrix} g_i(x_i) - a_i - S_i = 0 \\ g_i(x_i) - d_i - S_j = 0 \end{pmatrix} \text{ где } \begin{pmatrix} S_i \geq 0 \\ S_j \geq 0 \end{pmatrix} \quad (2.19 - 2.20)$$

Новые переменные S_i и S_j называются переменными типа СЛАКА, и эти переменные будем заменить логарифмическими барьерными членами в следующем этапе.

2. Замененные переменные типа СЛАКА (2.20) логарифмическими барьерными членами в целевой функции, где в результате замены получим новую целевую функцию с равенством ограничениям или условиям.

$$\left. \begin{array}{l} \text{Минимизация } J(x_i) - \mu \sum_{i=1}^m \log(S_i) \\ C_i(x_i) - S_i = 0, \quad i = 1 \dots m \end{array} \right\} \quad (2.21)$$

Здесь μ называется параметрами барьера имеет мелкое ненулевое значение, и C общая функции равенствах:

$$C_i(x_i) = \begin{bmatrix} g_i(x_i) - a_i \\ h_i(x_i) - b_i \end{bmatrix} \quad (2.22)$$

3. Включение ограничений равенства в целевую функцию с использованием множителей Лагранжа λ :

$$L(x, \lambda, S) = J(x) - \lambda^T (C(x) - s) - \mu \sum_{i=1}^m \log(S_i) \quad (2.23)$$

Задача оптимизации стало:

$$\text{Минимизация } L(x, \lambda, s) \quad (2.24)$$

4. Пишем условие Каруша–Куна–Таккера (ККТ) – производные первого порядка, которые являются необходимыми условиями в нелинейном программировании для оптимального решения.

$$\left. \begin{aligned} \nabla_x L(x, \lambda, s) &= \nabla_x J(x) - \nabla_x C(x)^T \lambda = 0 \\ \nabla_\lambda L(x, \lambda, s) &= C(x) - s = 0 \\ \nabla_s L(x, \lambda, s) &= -\mu s^{-1} e + \Lambda e = 0 \end{aligned} \right\} \quad (2.25)$$

Перепишем систему уравнений 2.25 в 2.26:

$$\left. \begin{aligned} \nabla_x J(x) - \nabla_x C(x)^T \lambda &= 0 \\ C(x) - s &= 0 \\ \Lambda s e &= \mu e \end{aligned} \right\} \quad (2.26)$$

Где e - вектор единицы $[1 \ 1 \ \dots \ 1]^T$.

5. Путём метода Ньютона вычисляем новые термы $\Delta x, \Delta \lambda$ и Δs в матричном форме:

$$\begin{bmatrix} H(x, y) & 0 & -A(x)^T \\ 0 & \Lambda & W \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -\nabla_x J(x) + \nabla_x C(x)^T \lambda \\ \mu e - \Lambda W e \\ -C(x) + s \end{bmatrix} \quad (2.27)$$

Здесь:

$$H(x, y) = \nabla_{xx}^2 L(x, \lambda, s) = \nabla_{xx}^2 J(x) - \sum_{i=1}^m \lambda_i \nabla_{xx}^2 C_i(x)^T \quad (2.28)$$

$$A(x) = -\nabla_x C(x) \quad (2.29)$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_m \end{bmatrix} (a), \quad W = \begin{bmatrix} S_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & S_m \end{bmatrix} (b) \quad (2.30)$$

6. Уравнение повторения:

$$\left. \begin{aligned} x^{k+1} &= x^k + \sigma^k \Delta x^k \\ \lambda^{k+1} &= \lambda^k + \sigma^k \Delta \lambda^k \\ s^{k+1} &= s^k + \sigma^k \Delta s^k \end{aligned} \right\} \quad (2.31)$$

Здесь: σ – является шагом повторения, и выбирается мелко хватит чтобы Двойная-Нормальная-Матрица $N(x, \lambda, s)$ будет положительно-определенный:

$$N(x, \lambda, s) = H(x, \lambda) + A(x)^T W^{-1} \Lambda A(x) \quad (2.32)$$

Если $N(x, \lambda, s)$ не положительно-определенный, тогда возмущаем матрикс Гессенца $H(x, \lambda)$:

$$\tilde{H}(x, \lambda) = H(x, \lambda) + \varepsilon I \quad (2.33)$$

Где: ε – мелкое значение.

Матрикс Гессенца является вторым условиям обеспечения оптимального значения, где содержит все производные уравнениях динамик второго порядка, и в результате расчёт должно имеет один знак положительный в задаче минимизации и отрицательный в задаче максимизации. Чтобы регулирует этот знак пользуем уравнение (2.33) или другие более сложности [18].

7. Условие схождения – максимальное абсолютное значения первые производные должны имеет мелкие значения меньше преопределённые значения $\epsilon_{\text{толерантность}}$.

$$\begin{aligned} \max |\nabla_x J(x) - \nabla_x C(x)^T \lambda| &\leq \epsilon_{\text{толер}} \\ \max |-\mu e + YWe| &\leq \epsilon_{\text{толер}} \\ \max |C(x) - S| &\leq \epsilon_{\text{толер}} \end{aligned} \quad (2.34)$$

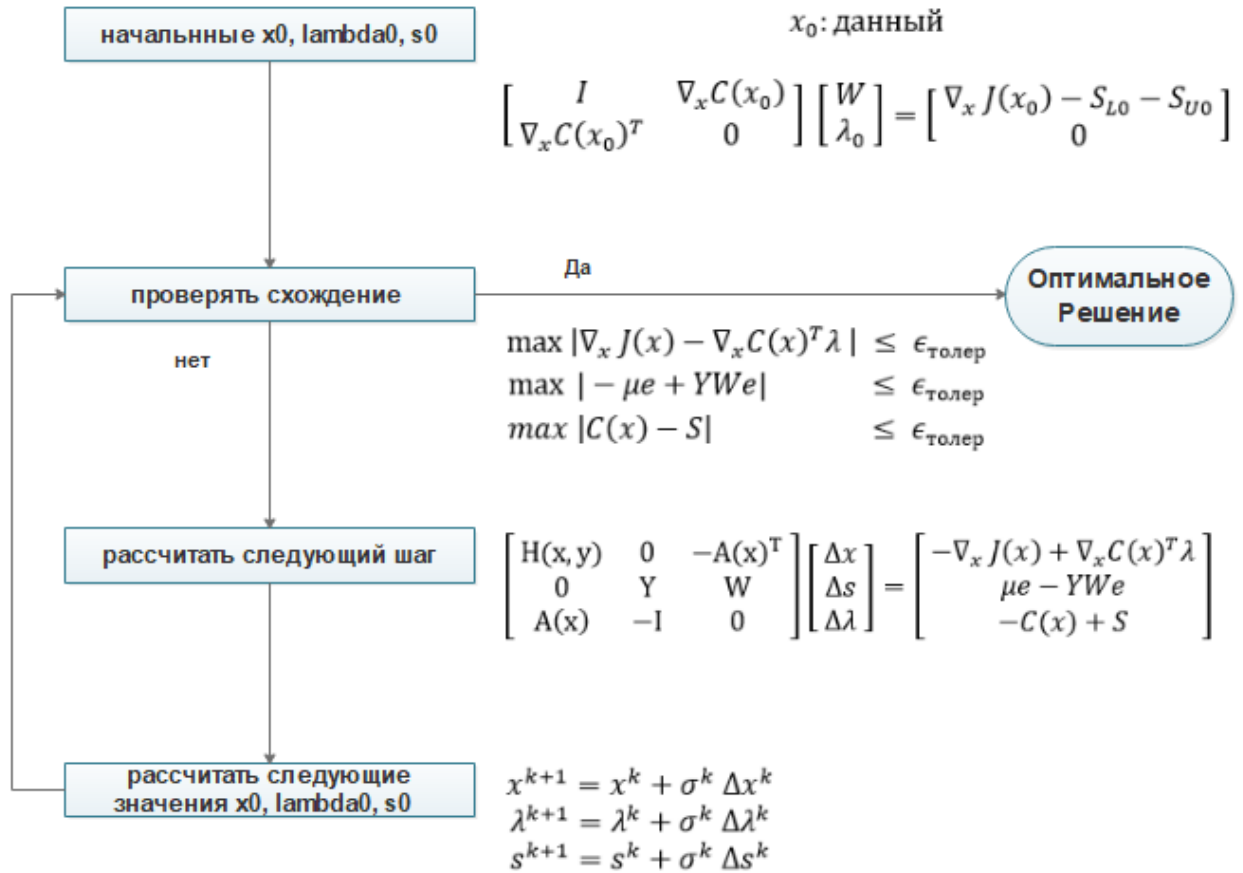
8. Начальные значения для параметров x, λ и s

x_0 : данный

$$\begin{bmatrix} I & \nabla_x C(x_0) \\ \nabla_x C(x_0)^T & 0 \end{bmatrix} \begin{bmatrix} W \\ \lambda_0 \end{bmatrix} = \begin{bmatrix} \nabla_x J(x_0) - S_{L0} - S_{U0} \\ 0 \end{bmatrix} \quad (2.35)$$

Закключаем логарифм MBT в следующем диаграмме 2.1.

Диаграмма 2.1. логарифм метода внутренней точки MBT



2.6.2 Обычные дифракционные уравнения

Обычные дифракционные уравнения решаются вместе с уравнением оптимизации (2.26) одновременно, т.е. необходимо преобразовать дифракционные уравнения в алгебраические уравнения. Для того необходимо брать метод дискретизации на базе численных метод, решавшие обычных дифракционных уравнений: метод Эйлера, Трапециевидный метод, метод Эрмита-Симпсона, метод Коллокации и другие. С помощью метода Эйлера пишем следующие приближительные уравнения динамики, [18]:

$$\dot{x} = f(x(t), u(t)) \Rightarrow \xi(x, u) = x_{k+1} - x_k - hf(x_k, u_k) = 0 \quad (2.36)$$

Расширенное уравнение Лагранжа (расширяем уравнение 2.23):

$$L(x, v, \lambda, s) = F(x, u) - v^T \xi(x, u) - \lambda^T (C(x, u) - s) - \mu \sum_{i=1}^m \log(s_i) \quad (2.37)$$

Где: v, λ являются множителем Лагранжа, $C(x, u)$ ограничения траектории и параметров динамики, s параметры типа СЛАКА, μ параметры барьера и имеет мелкое ненулевое значение, F целевая функция и имеет следующая форма:

$$J = \int_{t_i}^{t_f} \phi(x, u, t) dt \Rightarrow F(x_M) = \phi(x(t_f), t_f) = \phi(x_M);$$

$$\text{и } \dot{z} = \phi(x, u, t) \quad (2.38)$$

Пишем условие Каруша–Куна–Таккера (ККТ) – производные первого порядка для расширенной функции Лагранжа (2.37):

$$\left. \begin{aligned} \nabla_x L(x, \lambda, v, u, s) &= \nabla_x F(x, u) - \nabla_x \xi(x, u)^T v - \nabla_x C(x)^T \lambda = 0 \\ \nabla_\lambda L(x, \lambda, v, u, s) &= C(x) - s = 0 \\ \nabla_v L(x, \lambda, v, u, s) &= \xi(x) = 0 \\ \nabla_s L(x, \lambda, v, u, s) &= -\mu e + \Lambda s e = 0 \\ \nabla_u L(x, \lambda, v, u, s) &= \nabla_u F(x, u) - \nabla_u \xi(x, u)^T v - \lambda^T (\nabla_u C(x, u)) \end{aligned} \right\} \quad (2.39)$$

$$\nabla_{x_M} L(x, \lambda, v, s) = \nabla_{x_M} F(x_M, u) - v^T \nabla_{x_M} \xi_K(x, u) - \lambda^T \nabla_{x_M} C(x, u) = 0 \quad (2.40)$$

Уравнение (2.40) является условием трансверсальности.

С помощью метода Ньютона вычисляем новые термы $\Delta x, \Delta \lambda, \Delta v$ и Δs в матричном форме:

$$A X = B \quad (2.41)$$

$$\begin{aligned}
& \begin{bmatrix} H_x(x_k, u, \lambda, v) & H_x(x_{k+1}, u, \lambda, v) & A_x(x, u) & B_x(x, u) & 0.00 \\ -A_x(x_k, u)^T & -A_x(x_{k+1}, u)^T & 0.00 & 0.00 & -I \\ -B_x(x_k, u)^T & -B_x(x_{k+1}, u)^T & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & W & 0.00 & \Lambda \\ H_u(x_k, u, \lambda, v) & H_u(x_{k+1}, u, \lambda, v) & A_u(x, u) & B_u(x, u) & 0.00 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta x_{k+1} \\ \Delta \lambda \\ \Delta v \\ \Delta s \\ \Delta u \end{bmatrix} \\
& = \begin{bmatrix} -\nabla_x F(x_k, u) + \nabla_x \xi_K(x_k, u)^T v + \nabla_x C(x_k, u)^T \lambda \\ -\nabla_x F(x_{k+1}, u) + \nabla_x \xi_K(x_{k+1}, u)^T v + \nabla_x C(x_{k+1}, u)^T \lambda \\ -C(x, u) + s \\ -\xi_K(x, u) \\ \mu e - \Lambda s e \\ -\nabla_u F(x, u) + \nabla_u \xi_K(x, u)^T v + \nabla_u C(x, u)^T \lambda \end{bmatrix} \quad (2.42)
\end{aligned}$$

В матриксах A , W и Λ имеют формы (2.39), а H , A и B :

$$\begin{aligned}
H_x(x, u, \lambda, v) &= \nabla_{xx}^2 L(x, v, \lambda, s, u) \\
&= \nabla_{xx}^2 F(x, u) - v^T \nabla_{xx}^2 \xi_K(x, u) - \lambda^T \nabla_{xx}^2 C(x, u) \quad (2.43. a)
\end{aligned}$$

$$\begin{aligned}
H_u(x, u, \lambda, v) &= \nabla_{uu}^2 L(x, v, \lambda, s, u) \\
&= \nabla_{uu}^2 F(x, u) - v^T \nabla_{uu}^2 \xi_K(x, u) - \lambda^T \nabla_{uu}^2 C(x, u) \quad (2.43. b)
\end{aligned}$$

$$A_x(x, u) = -\nabla_x C(x, u)^T \quad (2.44. a)$$

$$A_u(x, u) = -\nabla_u C(x, u)^T \quad (2.44. b)$$

$$B_x(x, u) = -\nabla_x \xi(x, u)^T \quad (2.45. a)$$

$$B_u(x, u) = -\nabla_u \xi(x, u)^T \quad (2.45. b)$$

Уравнение повторения:

$$\left. \begin{aligned} x_{i+1}^k &= x_i^k + \sigma^k \Delta x^k \\ x_{i+1}^{k+1} &= x_i^{k+1} + \sigma^k \Delta x^{k+1} \\ \lambda^{k+1} &= \lambda^k + \sigma^k \Delta \lambda^k \\ v^{k+1} &= v^k + \sigma^k \Delta v^k \\ s^{k+1} &= s^k + \sigma^k \Delta s^k \\ u^{k+1} &= u^k + \sigma^k \Delta u^k \end{aligned} \right\} \quad (2.46)$$

Где: σ – является шагом повторения, и выбирается мелко хватит чтобы Двойная-Нормальная-Матрица $N(x, \lambda, s)$ будет положительно-определенный, уравнение (2.32). Если $N(x, \lambda, s)$ не положительно-определенный, тогда возмущаем матрикс Гессенца $H(x, \lambda)$, уравнение (2.33). Условие схождения – максимальное абсолютное значения первые производные должны имеет мелкие значения меньше преопределённые значения $\epsilon_{\text{толерантность}}$ и даётся в уравнении (2.34).

Логарифм решения новой расширенной системы похож на логарифм в диаграмме 2.1.

2.7 Оформление задачи оптимизации траектории полёта ГЛА

Данная задача оптимизации – это искать оптимальную траекторию полёта ГЛА, обеспечивающая минимальный расход топлива, при данных начальных и конечных координатах полёта, ограничениях параметров траектории и динамики. Перепишем уравнения движения в (2.47-2.50).

$$\begin{aligned}\frac{dV}{dt} &= \frac{T_x - D}{m} - g \sin(\gamma) + V_{om} \\ \frac{d\gamma}{dt} &= \frac{1}{V} \left(\frac{L \cos(\mu_a) - Y_a \sin(\mu_a)}{m} - g \cos(\gamma) + \gamma_{om} \right) \quad (2.47) \\ \frac{d\chi}{dt} &= \frac{1}{V} \left(\frac{L \sin(\mu_a) + Y_a \cos(\mu_a)}{m \cos(\gamma)} + \chi_{om} \right)\end{aligned}$$

$$\begin{aligned}\dot{h} &= V \sin(\gamma) \\ \dot{\theta} &= V \cos(\gamma) \cos(\chi) / (R \cos(\phi)) \quad (2.48) \\ \dot{\phi} &= V \cos(\gamma) \sin(\chi) / R\end{aligned}$$

$$\dot{m} = -\nu = \nu_{\text{ном}} + (\nu_{\text{макс}} - \nu_{\text{ном}}) * \eta_t \quad (2.49)$$

$$\begin{aligned}V_{om} &= \omega^2 R \cos^2(\phi) (\sin(\gamma) - \cos(\gamma) \tan(\phi) \sin(\chi)) \\ \gamma_{om} &= \frac{V^2}{R} \cos(\gamma) + 2\omega V \cos(\phi) \cos(\chi) + \omega^2 R \cos^2(\phi) (\cos(\gamma) + \sin(\gamma) \tan(\phi) \sin(\chi)) \quad (2.50) \\ \chi_{om} &= -\frac{V^2}{R} \cos(\gamma) \cos(\chi) \tan(\phi) + 2\omega V (\tan(\gamma) \cos(\phi) \sin(\chi) - \sin(\phi)) - \frac{\omega^2 R}{\cos(\gamma)} \sin(\phi) \cos(\phi) \cos(\chi)\end{aligned}$$

Термины V_{om} , γ_{om} и χ_{om} возникают при учёте вращения земли вокруг вертикальной оси Z. Уравнения аэродинамики поставили с характеристиками самолёта в главе 3.

Параметры управления движения является: α – угол атаки, μ_a – угол крена в системе координата траектории, η_t - положение дроссельного клапана [0-1], т.е. $u = \{\alpha, \mu_a, \eta_t\}$, и можно смотреть влиянию β – угол скольжения на результатов, т.е. $u = \{\alpha, \beta, \mu_a, \eta_t\}$. Эти параметры будем определить их значения в процессе оптимизации, чтобы достигать минимизацию конечного значения топлива.

Ограничения параметров движения и траектории поставили в таблице 2.1.

Таблица 2.1 ограничения параметров движения

Параметр	Мин. Значение	Макс. Значение	Единица
$R = R_E + h$	30000	30000	М
γ	-89 (-1.55)	89 (1.55)	Градус (Рад.)
χ	-180 (-3.14)	180 (3.14)	Градус (Рад.)
μ_a	-5 (-0.087)	5 (0.087)	Градус (Рад.)
θ'	-89 (1.55)	89 (1.55)	Градус (Рад.)
ϕ'	-45 (-0.78)	45 (0.78)	Градус (Рад.)
ψ'	-180 (-3.14)	180 (3.14)	Градус (Рад.)
V	5	8	М
m (Масс)	54,431.07	136,077.7	Кг
$\dot{m}_{\text{топ}}$	9.071	163.29	Кг/с
α	-10 (-0.17)	10 (0.17)	Градус (Рад.)
β	-10 (-0.17)	10 (0.17)	Градус (Рад.)

Для начальных и конечных координат посмотрим две траектории, первая траектория будем называться **АБ**, а **ВГ** вторая траектория, где **А** и **В** является начальной точкой полёта, а **Б** и **Г** является конечной точкой полёта. Координаты этих точек поставили в таблице 2.2.

Таблица 2.2 начальные и конечные координаты полёта

	Широта ϕ° (Рад)	Долгота θ° (Рад)	Высота h [km]
точка А	55.58 (0.97)	37.90 (0.66)	30
точка Б	40.70 (0.71)	-73.70 (-1.28)	30
точка В	-13.05 (-0.23)	-45.25 (0.79)	30
точка Г	-18.19 (-0.27)	130.54 (2.28)	30

Уравнения (2.47-2.50) можно легко переписать в приблизительном виде (3.36), а ограничения в таблицах 2.1 и 2.2 перепишем в виде 2.19 и 2.20, и получим следующие уравнения:

$$\begin{aligned}
 +R - 30000 - s_1 &= 0 \quad (2.51), & +\gamma - 1.55 - s_3 &= 0 \quad (2.52), & +\chi - 3.14 - s_5 &= 0 \quad (2.53), \\
 -R + 30000 - s_2 &= 0 \quad (2.51), & -\gamma + 1.55 - s_4 &= 0 \quad (2.52), & -\chi + 3.14 - s_6 &= 0 \quad (2.53), \\
 +\mu_a - 0.087 - s_7 &= 0 \quad (2.54), & +V - 5 - s_9 &= 0 \quad (2.55), & +m - 54,431.07 - s_{11} &= 0 \quad (2.56), \\
 -\mu_a + 0.087 - s_8 &= 0 \quad (2.54), & -V + 8 - s_{10} &= 0 \quad (2.55), & -m + 136,077.7 - s_{12} &= 0 \quad (2.56), \\
 +\alpha - 0.17 - s_{13} &= 0 \quad (2.57), & +\beta - 0.17 - s_{15} &= 0 \quad (2.58), & +\phi - A_\phi - s_{17} &= 0 \quad (2.59), \\
 -\alpha + 0.17 - s_{14} &= 0 \quad (2.57), & -\beta + 0.17 - s_{16} &= 0 \quad (2.58), & -\phi + B_\phi - s_{18} &= 0 \quad (2.59), \\
 +\theta - A_\theta - s_{19} &= 0 \quad (2.60), & & & & \\
 -\theta - B_\theta - s_{20} &= 0 \quad (2.60), & & & &
 \end{aligned}$$

Где s_i имеет положительное значение:

$$s_i \geq 0, i = 1, 2 \dots 20 \quad (2.61)$$

Наша целевая функция – это конечная масса топлива или самолёта, где масса топлива должна минимизироваться, а масса самолёта максимизироваться, поставим минус в целевой функции, чтобы перевернуть задачу от минимизации в максимизацию:

$$F(x) = -m_{\text{кон. сам.}} \quad (2.62)$$

Следующий этап является объединением уравнений ограничений (2.51-2.60), динамик (2.47-2.50) и целевой функции (2.62) в функции Лагранжа имеющая форма (2.37). Результирующее уравнение поставили в приложении Б, уравнение Б.1. в уравнении (Б.1), m число ограничения и равно 20, уравнения (2.51-2.60).

Матрикс A в уравнении (2.42) имеет размер $n_{xk} + n_{xk+1} + n_{\lambda} + n_v + n_s + n_u = 64$ переменных \times 64 уравнений. Где $n_{xk} = 7$ и $n_{xk+1} = 7$ – число параметров динамики. $n_{\lambda} = 20$ – число первых множителей Лагранжа. $n_v = 7$ – число вторых множителей Лагранжа. $n_s = 20$ – число дополнительных параметров. $n_u = 3$ – число параметров управления. Такой большой матрикс порождаем сразу в МАТЛАБ. Все коды МАТЛАБ поставили в приложении В.

Глава 3 Характеристики изучаемого самолёта

3.1 Описание ГЛА пользующий в диссертации

Общий пример гиперзвуковой аэродинамической модели (GHAME) был предоставлен Исследовательским Центром Компьютерного Моделирования НАСА, [17]. Модель состоит из реалистичных данных аэродинамических коэффициентов в режиме гиперзвукового полета. Данные были представлены с целью предоставления имитационной модели для анализа исследований и разработок.

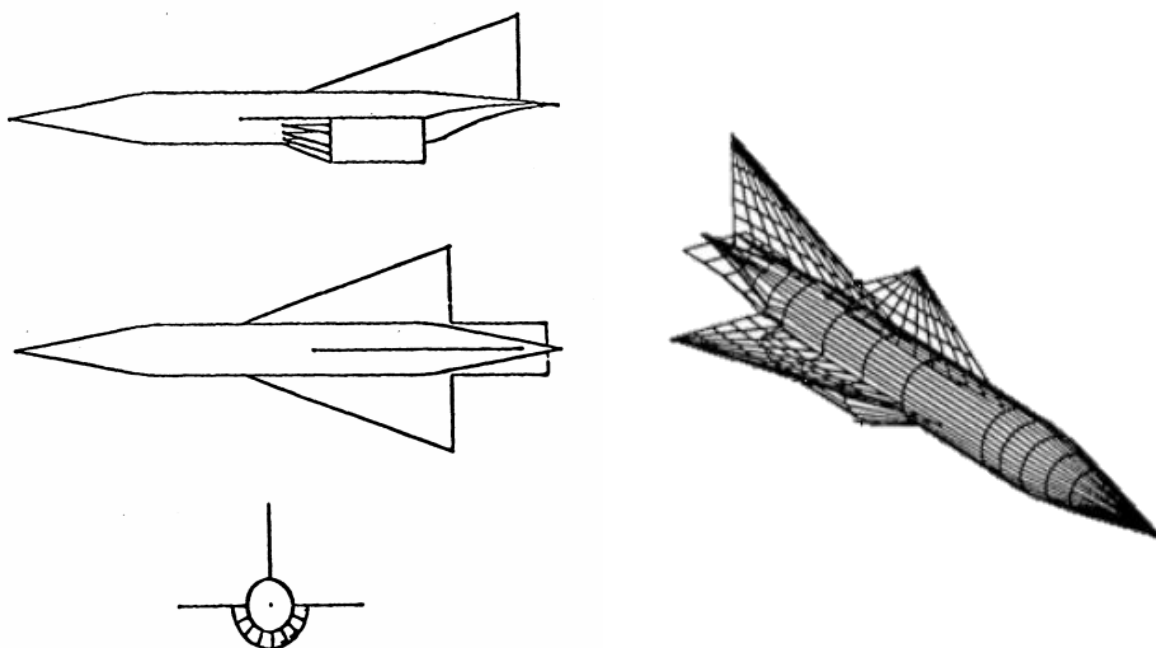


Рисунок 3.1 Конфигурации GHAME

Геометрия GHAME построена из простых геометрических фигур и показана на рисунке 3.1. Это позволило упростить оценку массовых характеристик. Основная структура была смоделирована в виде цилиндра диаметром 20 футов (6.09 м) и длиной 120 футов (36.57 м). Это обеспечило внутренний объем, необходимый для хранения жидкого топлива водород. Пара конусов с половинным углом 10 градусов была прикреплена к этому цилиндру для того чтобы сформировать нос и хвост. Крылья и вертикальный хвост были смоделированы в виде тонких треугольных пластинок. Крылья-начинаются в

средней точке фюзеляжа и не имеют двугранного углы. Модуль двигателя оборачивается вокруг нижней поверхности фюзеляжа. Аэродинамические параметры конструкции: опорная площадь составляет 6000 квадратных футов (557.42 м^2), опорная хорда 75 футов (22.86 м), а размах находится в 80 футов (24.38 м).

Таблица 3.1 Справочные данные GNAME

опорная площадь	557.42 м^2
опорная хорда	22.86 м
размах	24.38 м
длина	71.14 м
Взлетный вес	136,077.7 кг
Вес топлива	81,646.63 кг
Номинальный расход топлива	9.071 кг/с
Максимальный расход топлива	163.29 кг/с

3.2 Массовые свойства

Массовые свойства GNAME имеет тот же порядок величины, что и текущий сверхзвуковой круизный самолет, и были специально получены из самолета ХВ-70. Взлетный вес был смоделирован, чтобы быть 300,000 фунтов ($136,077.7 \text{ кг}$) с 60% (180,000 фунтов ($81,646.63 \text{ кг}$)) в качестве жидкого водородного топлива. Номинальный справочный центр массы находится в точке $0.33 c_{mac}$. Массовые моменты инерции были рассчитаны на основе простых геометрических форм и приложены в таблице 3.2.

Таблица 3.2 Массовые моменты GNAME

На взлете	При выгорании топлива
$I_x = 1572748.82 \text{ кг. м}^2$	$I_x = 1179561.61 \text{ кг. м}^2$
$I_y = 31590558.23 \text{ кг. м}^2$	$I_y = 19252614.89 \text{ кг. м}^2$
$I_z = 32539630.8 \text{ кг. м}^2$	$I_z = 20201687.45 \text{ кг. м}^2$

$I_{xz} = 379629.026 \text{ кг. м}^2$	$I_{xz} = 379629.026 \text{ кг. м}^2$
---------------------------------------	---------------------------------------

3.3 Аэродинамические данные

Данные GHAME состоят из таблиц аэродинамических коэффициентов, используемых для расчета сил и моментов на самолете в целях моделирования. Каждый коэффициент зависит от угла атаки. Строки организованы в зависимости от числа Маха, а колонки в зависимости от угла атаки в градусах [11,12]. Все данные рисовали в зависимости от угла атаки и число Маха в графах (3.2 - 3.8).

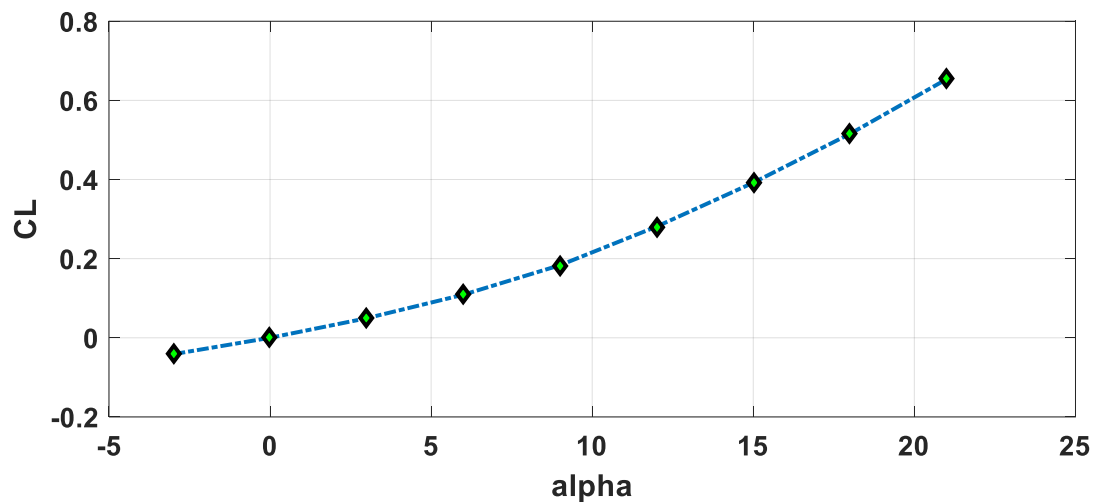


Рисунок 3.2 зависимости коэффициента подъемной силы от угла атаки

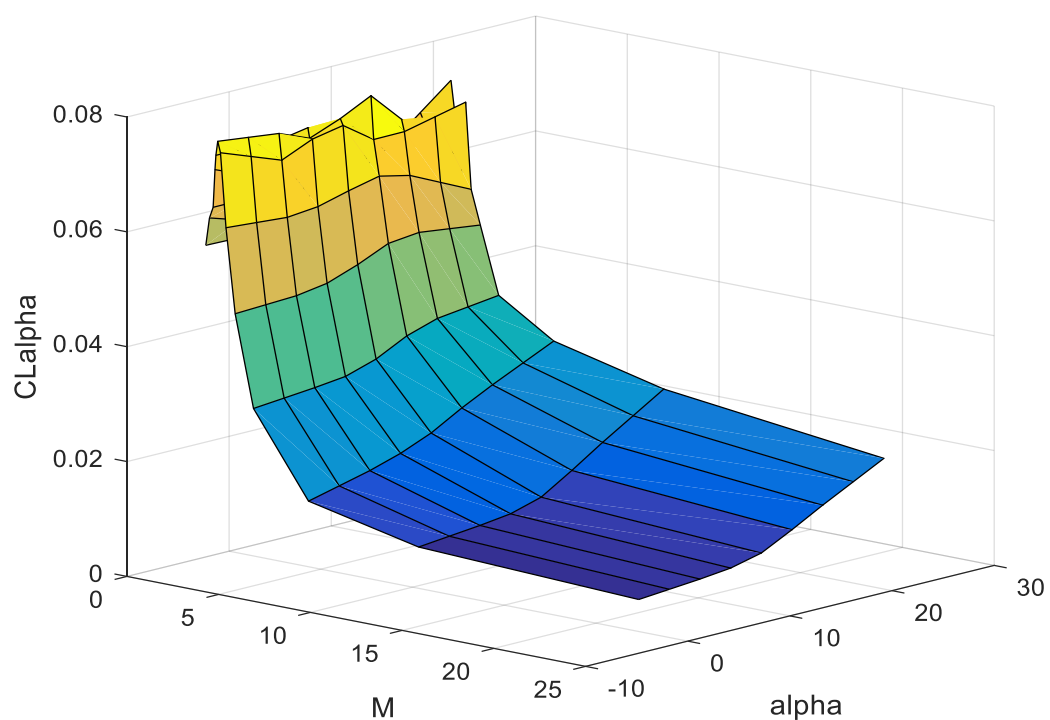


Рисунок 3.3 зависимости CL_{α} от угла атаки и числа Маха

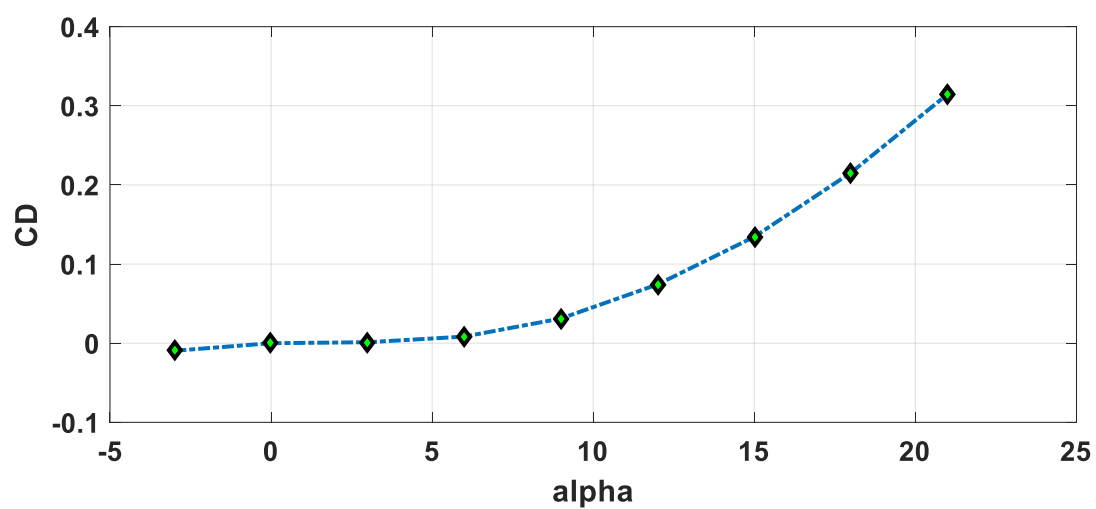


Рисунок 3.4 зависимости коэффициента силы сопротивления от угла атаки

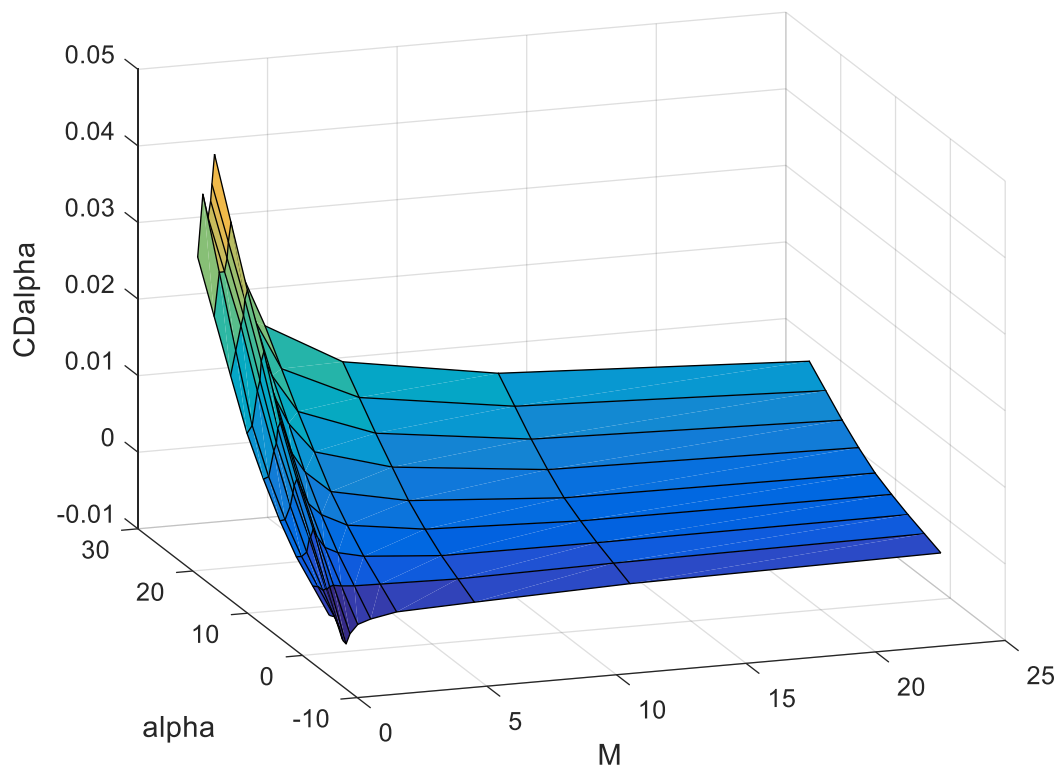


Рисунок 3.5 зависимости CD_{α} от угла атаки и числа Маха

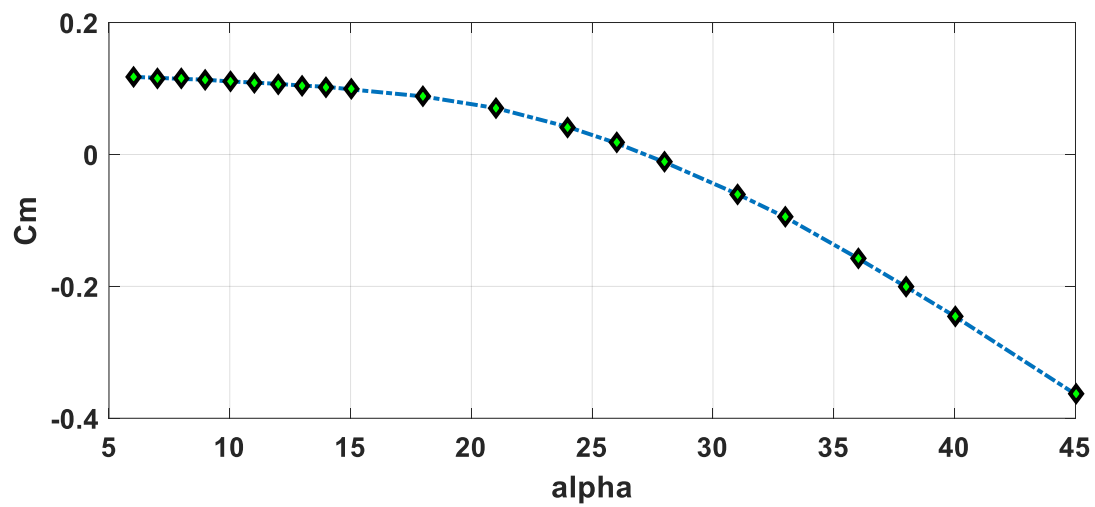


Рисунок 3.6 зависимости коэффициента момента тангажа C_m от угла атаки

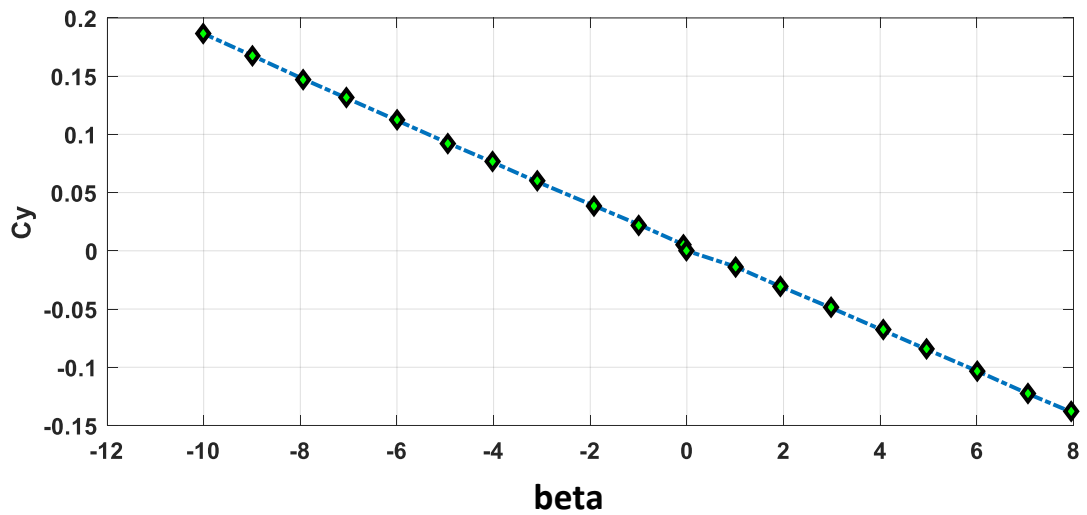


Рисунок 3.7 зависимости коэффициента боковой силы C_y от угла скольжения

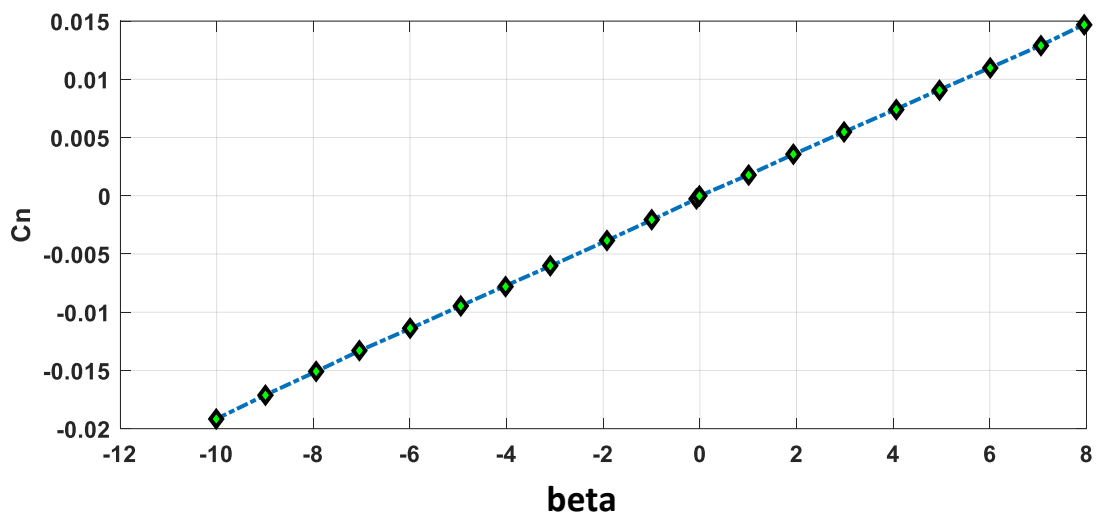


Рисунок 3.8 зависимости коэффициента момента рыскания C_n от угла скольжения

Полиномы и графы этих данных порождали в МАТЛАБ где он сразу перебрасывает данные таблицы в графы (3.2 – 3.8) на базе преопределенной степени полинома. Графы (3.2 – 3.8) рисовали на базе полинома второй степени.

Глава 4 Результаты и их анализ

4.1 Программ управления $u = \{\alpha, \mu_a, \eta_t\}$

В этой части посмотрим результаты оптимизации разных траектории, переопределяли в главе 2. Главные входы кода МАТЛАБ эти место начала и конца полёта, время полёта, который можно считать приблизительно. Выходы кода эти скорость и высота полёта, углы наклона траектории вертикальный и горизонтальный, масса самолёта, положение дроссельного клапана, угол атаки и крена, широта и долгота траектории.

Перепишем начальное и конечное значение траектории в таблице 4.1. Где первая траектория АБ, а вторая траектория ВГ.

Таблица 4.1 начальные и конечные координаты траектории

	Широта ϕ° (Рад)	Долгота θ° (Рад)	Высота h [km]
точка А	55.58 (0.97)	37.90 (0.66)	30
точка Б	40.70 (0.71)	-73.70 (-1.28)	30
точка В	-13.05 (-0.23)	-45.25 (0.79)	30
точка Г	-18.19 (-0.27)	130.54 (2.28)	30

4.1.1 Результаты первой траекторий АБ

Точка начально полета первой траекторий $(\phi^\circ, \theta^\circ, h) = (55.58, 37.90, 30)$, а точка конца полёта $(\phi^\circ, \theta^\circ, h) = (40.70, -73.70, 30)$. Приблизительное значение этих точек можно получить из карты Google и тоже приблизительно расстояние между этими точками, которое является 8277 км, т.е. приблизительно время полёта является 4523 сек (1.25 час). Это время нужно в качестве входа кода, если он не обхватывает повышаем этот значение, и наоборот.

Эта профиль считается длинный, но существуют обычные самолёты (не ГЛА) сможет летать прямо без посадки, но время полёта большое около 9-11 часов.

Высота полёта фиксировали на 30000 метров, а скорость имеет диапазон [5-6] Мах, где 5 минимальная скорость для гиперзвуковых самолётов. Из рис. 4.1 сможем заметить, что самолёт полетела на этой профили с одной скоростью 6 Мах (1830 м/с). Это обеспечивает равномерное неускоренное движение по маршруте.

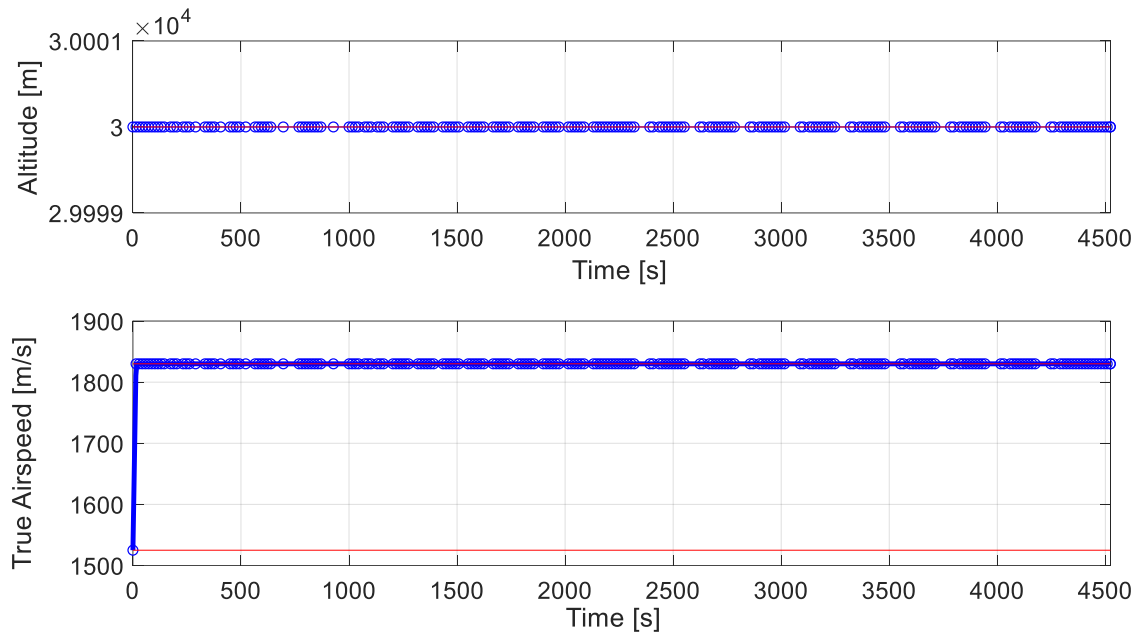


Рисунок 4.1 высота и скорость самолёта первой профили

В рисунке 4.1, верхний граф является высотой, а нижний скоростью полёта. В рисунке 4.2 показано параметры управления: угол крена в системе координаты траектории (верхний граф) и положение дроссельного клапана (нижний граф). Угол крена ограничили в диапазоне $[-1 - 1]$ градус. А диапазон положения дроссельного клапана $[0 - 1]$. Из первого графа замечаем, что угол крена получила на этой профили значение равно 0.5 градуса, а в конце траектории значение равно 0.0 градуса такой же и в начале траектории, грубо-скажем, что она не будет ощущено для тех сидящие в самолёте.

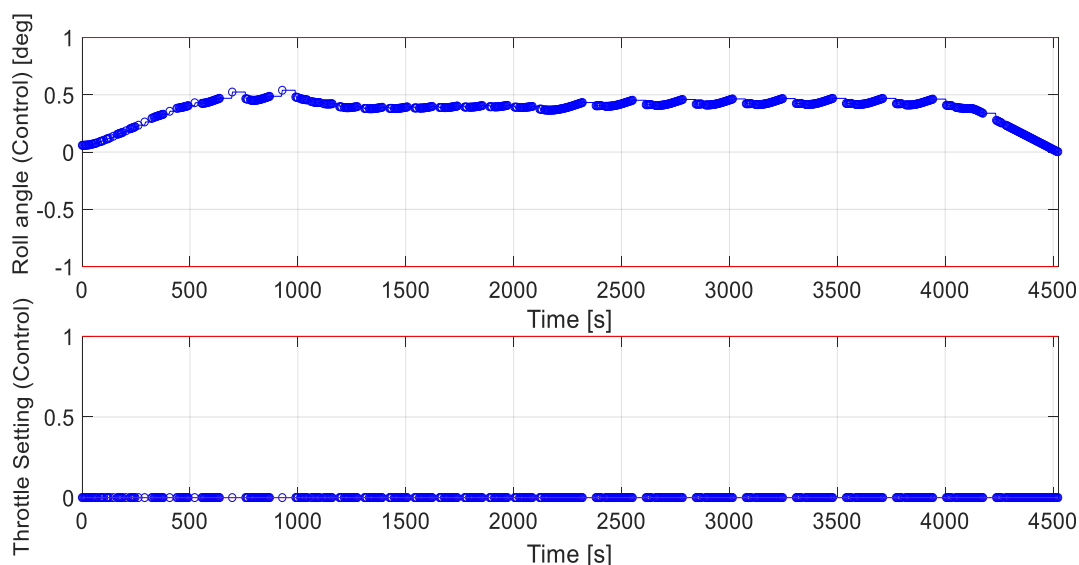


Рисунок 4.2 угол крена и положение дроссельного клапана первой профили

Из второго графа рис. 4.2, заметим, что положение дроссельного клапана получил значение нули, это значит, что самолёт летает по номинальному расход топлива см. уравнение 1.14, который равно 9.071 кг/с, см. таблицу 3.1.

Рисунок 4.3 показывает изменение массы самолёта и углы атаки. Масса самолёта равно массе топлива и массе самолёта без топлива, масса самолёта полная равно 136.077 тон, и масса топлива 81.646 тон, см. таблицу 3.1. угла атаки ограничила в диапазоне [-10 – 10] градус. Верхний граф рис. 4.3 показывает массу самолёта в килограммах, и сможем считать, что потребляемая масса топлива равно 41027 кг. Нижний граф рис. 4.3 показывает углу атаки, которая имеет значение равно 0.7 градус. Из аэродинамических данных сможем оценить значения углы атаки; снижение углы атаки снижает сопротивлению силу, и наоборот повышение углу атаку повышает сопротивлению силу. Запомним, что угол атак является параметром управления.

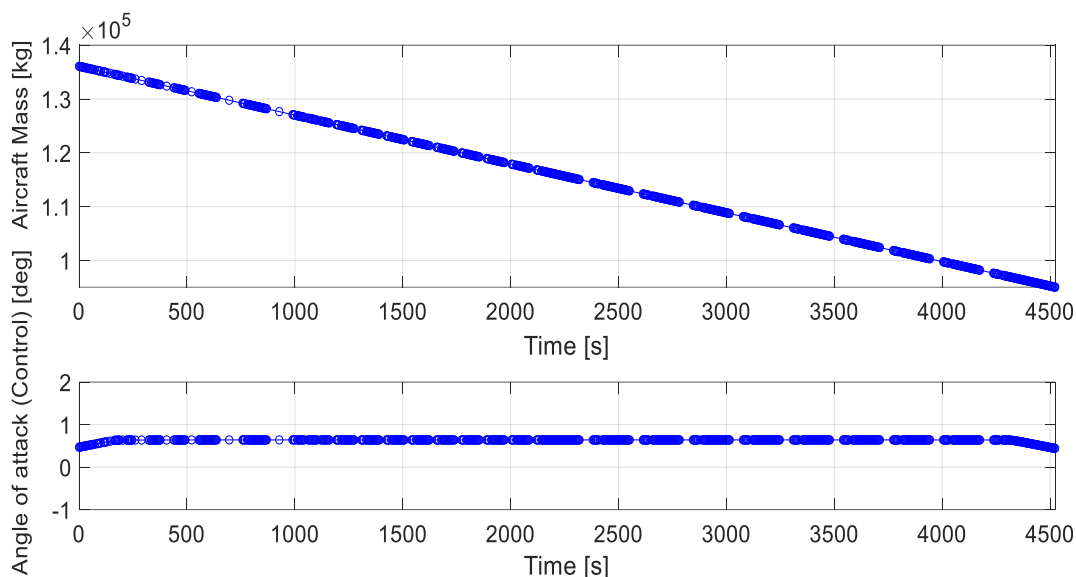


Рисунок 4.3 масса самолёта и угол атака первой профили

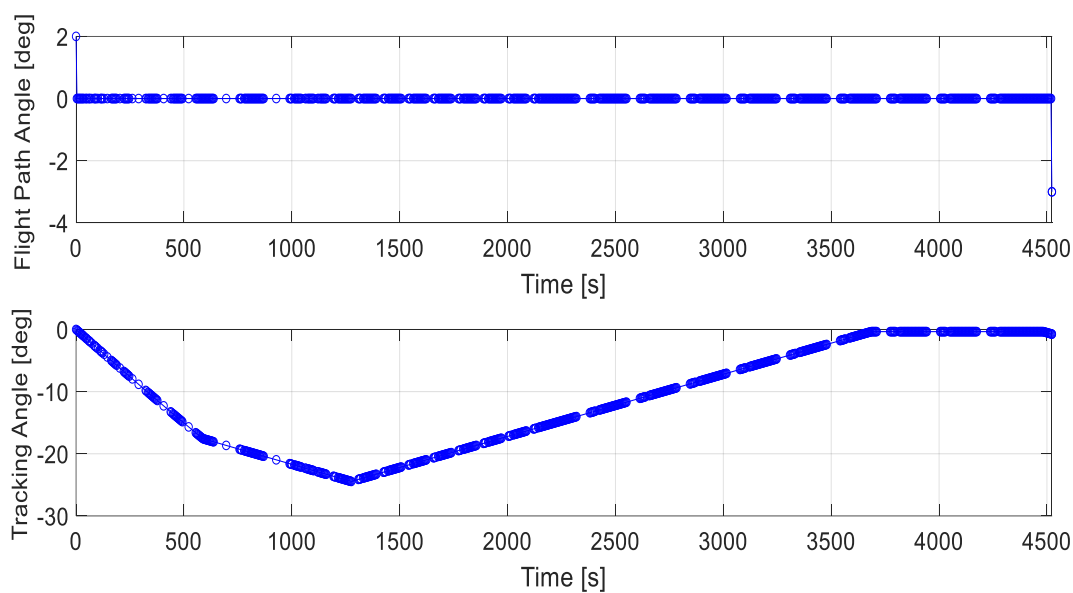


Рисунок 4.4 углы наклонения траектории вертикальная и горизонтальная первой профили

Углы наклонения траектории вертикальный и горизонтальный в градусах показаны в рисунке 4.4. Угол вертикального наклонения ограничено в диапазоне $[-89 - 89]$ градус, а угол горизонтального наклонения ограничено в диапазоне $[-180 - 180]$ градус. Верхний граф рис 4.4, показывает вертикальное наклонение, и его значение равно нулю, а горизонтальное наклонение, нижний граф рис 4.4, имеет разные значения и в конце стал равном нулю как ожиданно.

Рисунок 4.5 показывает углы широта и долгота в градусах. Угол широта показано в верхнем графе, сможем заметить, что начальное и конечное значение этого угла равны данным значениям этой траектории 55.58 и 37.90 градусов. В нижнем графе показано угол долготы с ожидаемым начальным и конечным значениями 40.70 и -73.70 градусов.

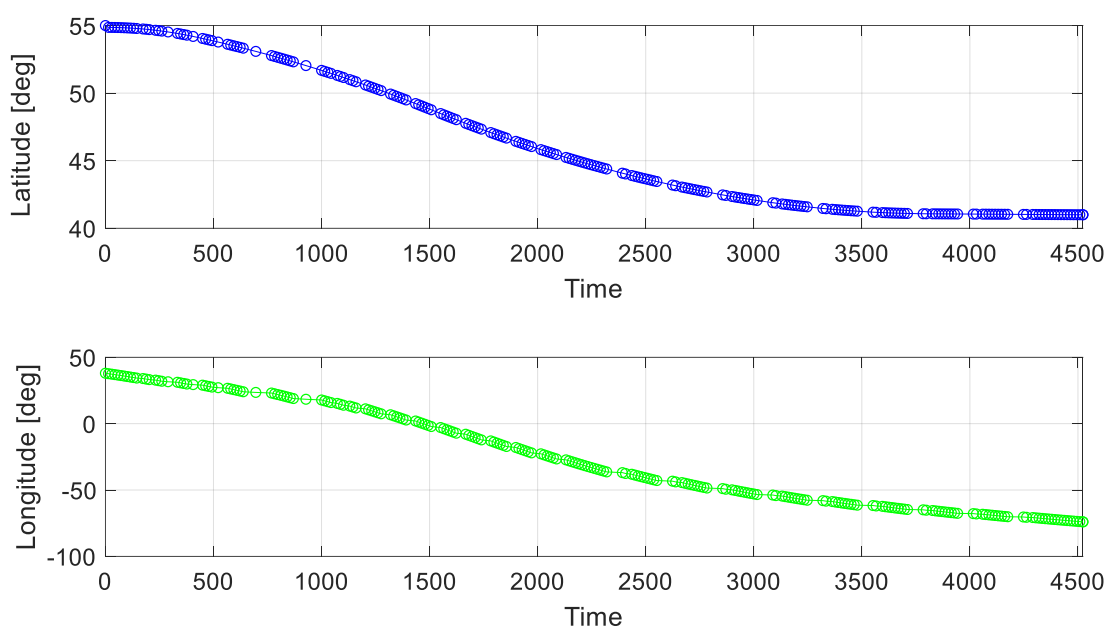


Рисунок 4.5 углы широты и долготы траектории первой профили

Углы широта и долгота, сможем рисовать их на карте земли, и результат будет траектория полета из точки А (55.58, 37.90, 30) в точку В (40.70, -73.70, 30). Этот траектории показано в рисунке 4.6 в красных точках, а точки начало и конце в чёрных точках. Вертикальная ось является широтой, а горизонтальной является долготой.

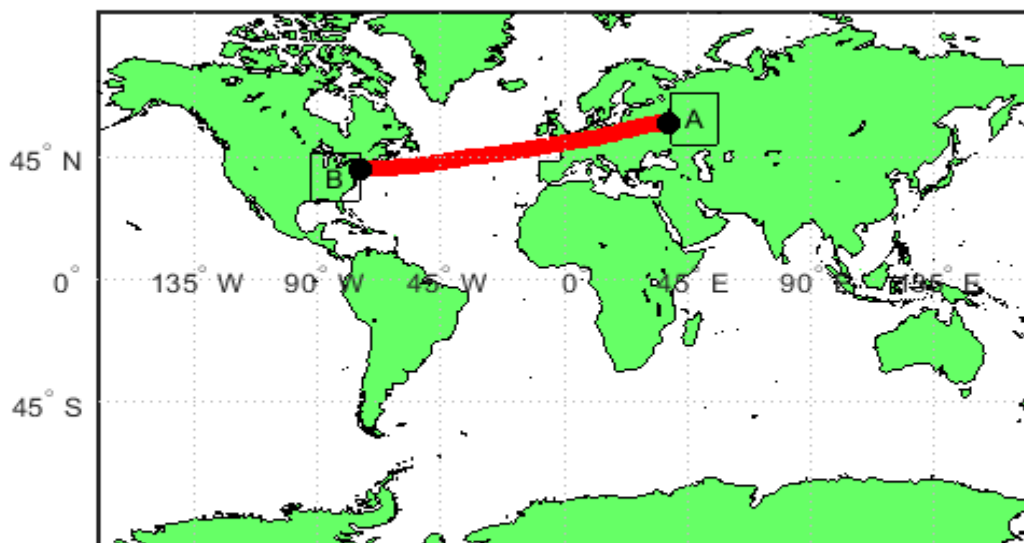


Рисунок 4.6 траектория полёта на карте земли – первая профиль

4.1.2 Результаты второй траекторий ВГ

Точка начально полета второй траекторий $(\phi^\circ, \theta^\circ, h) = (-13.05, -45.25, 30)$, а точка конца полёта $(\phi^\circ, \theta^\circ, h) = (-18.19, 130.54, 30)$. Приблизительное значение этих точек можно получить тоже из карты Google. Приблизительное расстояние между этими точками является 15,590 км, т.е. приблизительное время полёта является 8519 сек (2.36 час). Это время нужно в качестве входа кода, если он не обхватывает повышаем этот значение, и наоборот. Этот профиль полёта два раза длиннее первой профили и является один из самых длинных профилей, существующие в этой время. Обычные самолёты летает такие профили на нескольких посадок (не прямой полёт), где он связывает между Бразилия и Австралия.

Ограничения всех параметров движения в этой траектории такой же и в первой траектории. Высоту в этой траектории тоже фиксировали на 30 км,

показана в верхнем графе рис. 4.7, а скорость в нижнем графе. Скорость получила величину 6 маха (1830 м/с).

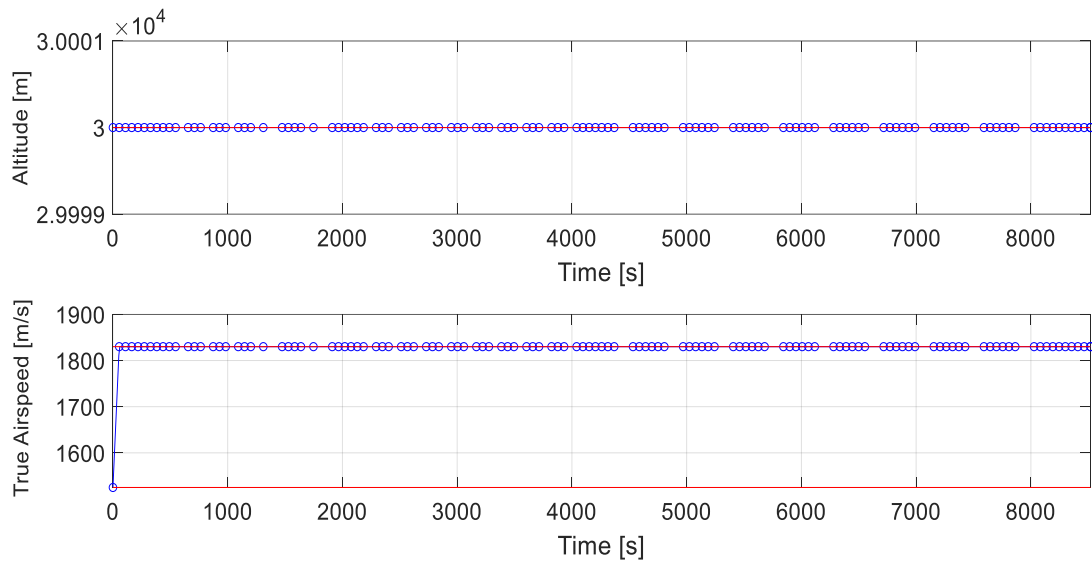


Рисунок 4.7 высота и скорость самолёта второй профили

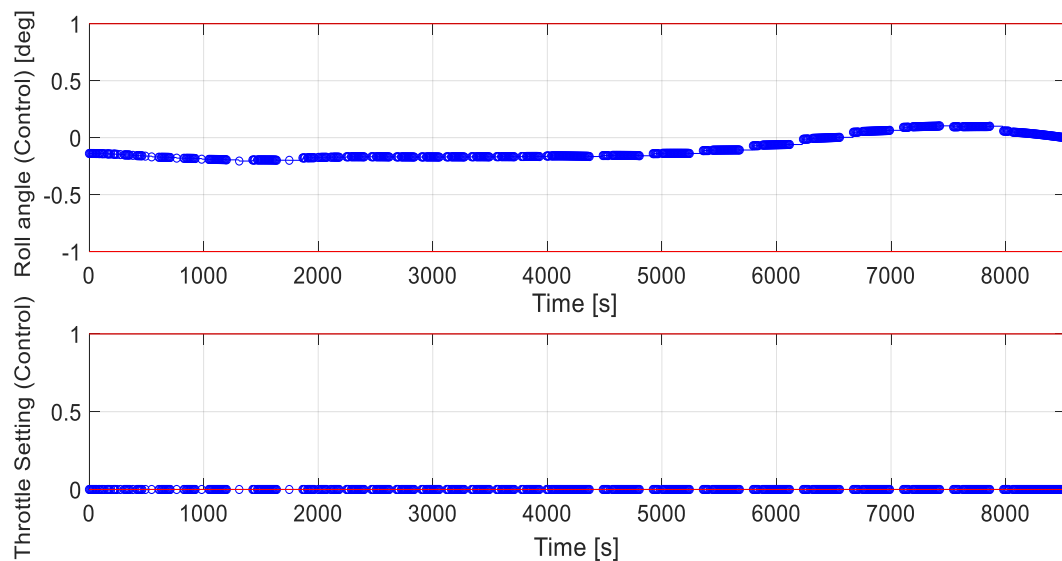


Рисунок 4.8 угол крена и положение дроссельного клапана второй профили

угол крена и положение дроссельного клапана показаны в рисунке 4.8. в верхнем графе сможем заметить, что величина угол крена получила разные мелкие значения, но в конце траектории равно нулю. Положение дроссельного клапана равно нулю, т.е. самолёт летает по номинальному расходу топлива.

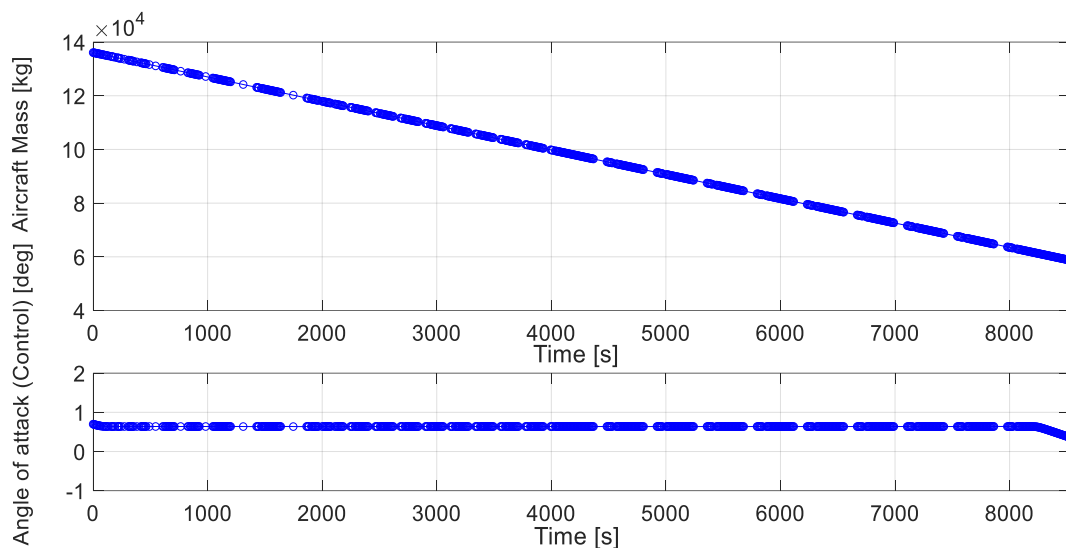


Рисунок 4.9 масса самолёта и угол атаки второй профили

Рисунок 4.9 показывает изменение массы самолёта и углы атаки. Верхний граф рис. 4.9 показывает массу самолёта в килограммах, и сможем считать, что потребляемая масса топлива равно 77277 кг. Нижний граф рис. 4.3 показывает углу атаки, которая имеет значение равно 0.7 градус. Из аэродинамических данных сможем оценить значения углы атаки; снижение углы атаки снижает сопротивлению силу, и наоборот повышение углу атаку повышает сопротивлению силу. Запомним, что угол атак является параметром управления.

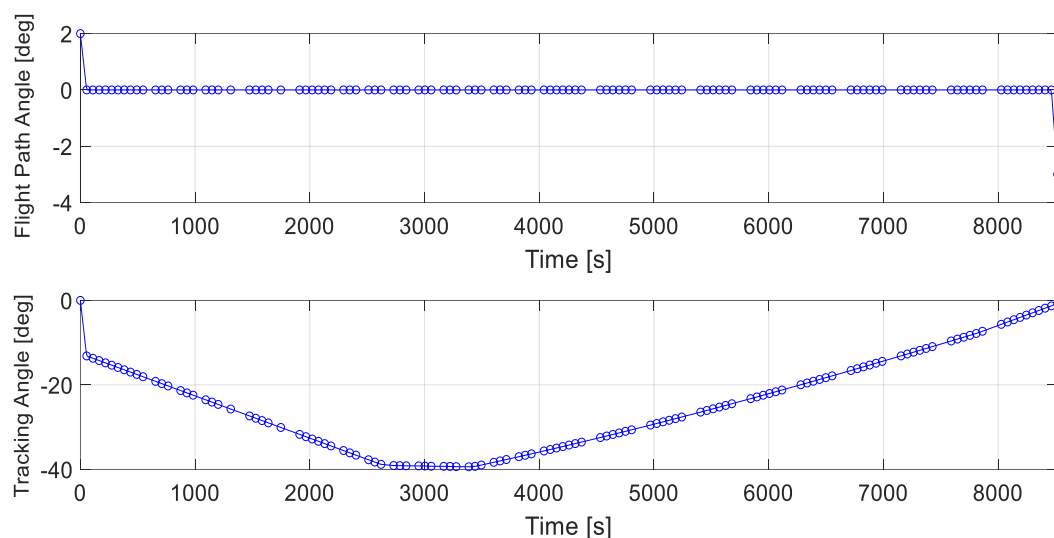


Рисунок 4.10 углы наклона траектории вертикальная и горизонтальная второй профили

Углы наклона траектории вертикальный и горизонтальный в градусах показаны в рисунке 4.10. Верхний граф показывает вертикальное наклонение, и его значение равно нулю, а горизонтальное наклонение, см. нижний граф рис 4.10, имеет разные значения и в конце стал равном нулю как ожидално.

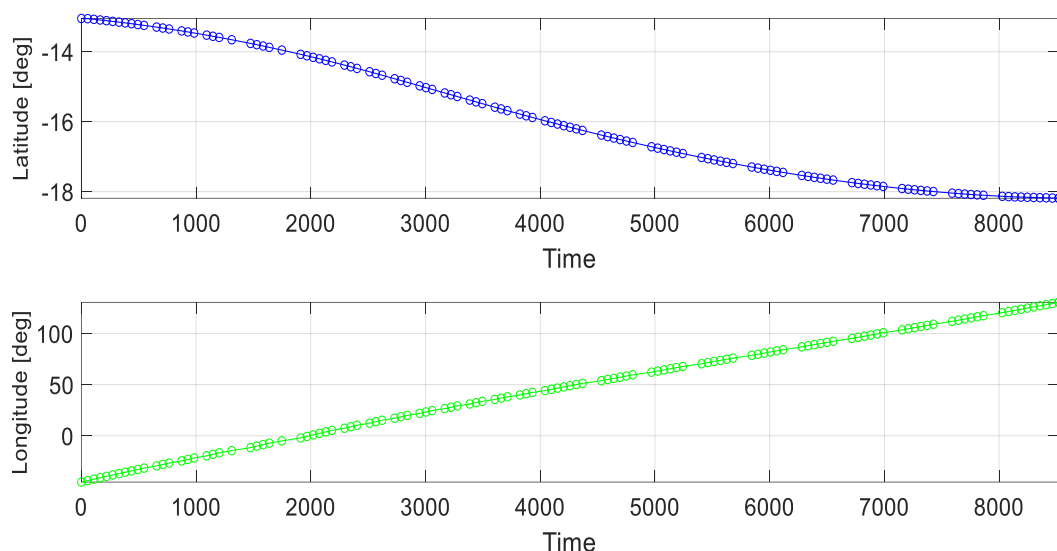


Рисунок 4.11 углы широты и долготы траектории второй профили

Рисунок 4.11 показывает углы широта и долгота в градусах. Угол широта показано в верхнем графе, и сможем заметить, что начальное и конечное значение этого угла равны данным значением этой траектории -13.05 и -18.19 градусов. В нижнем графе показано угол долготы с ожидаемым начальным и конечным значениями и -45.25 и 130.54 градусов.

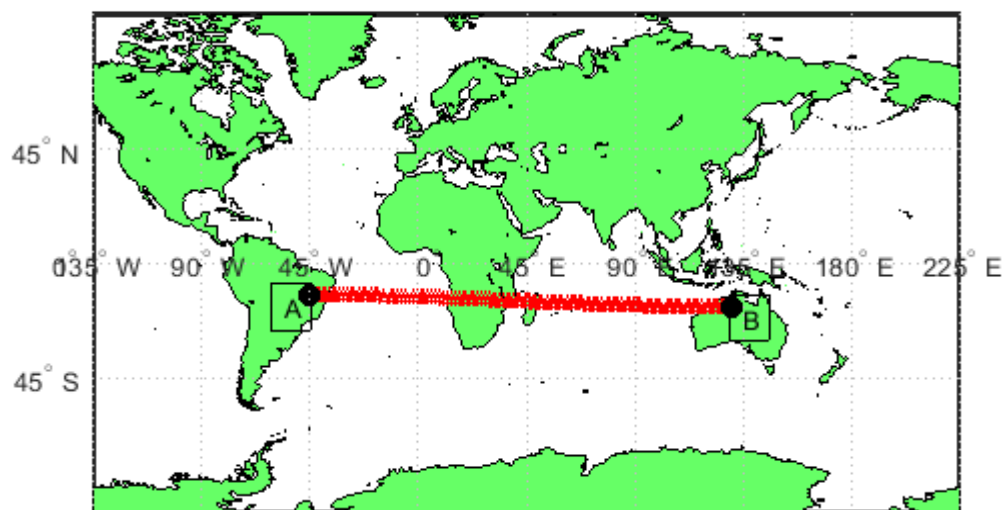


Рисунок 4.12 траектория полёта на карте земли - вторая профиль

Углы широта и долгота, сможем рисовать их на карте земли, и результат будет траектория полета из точки А (-13.05, -45.25, 30) в точку В (-18.19, 130.54, 30). Этот траектории показано в рисунке 4.12 в красных точках, а точки начало и конце в чёрных точках. Вертикальная ось является широтой, а горизонтальной является долготой.

Из результатов этих двух траектории сможем заметить, что параметры, действующие на вертикальном движении (угол атака, вертикальная наклонения углы траектории) получили фиксированные значения, потому что высота была фиксирована, а параметры, действующие на горизонтальном движении, получили разные значения, чтобы ведет самолёт в назначенные координаты.

4.2 Программ управления $u = \{\alpha, \beta, \mu_a, \eta_t\}$

В этой часть добиваем новый параметр управления, и посмотрим его закон управления и как аффекирует на других параметрах движения и управления.

Выводим двух траектории, которые же посмотрели в части 4.1, и поставим в сравнении результаты сравнения между параметрами без угла скольжения (no side-slip) и параметрами с углом скольжения (with side-slip). Не все параметры показали изменения при наличии угла скольжения, поэтому поставили графы только те параметры, которые подвергали изменение в их величинах.

4.2.1 Результаты второй траекторий АБ

Изменявшиеся параметры являются угол крена, угол атаки и угол скольжения сам. В рисунке 4.13 показали сравнение угол крена траектории при наличии действия угла скольжения (красная кривая) и без наличия действия угла скольжения (синяя кривая). Сможем заметить, что при наличии угла скольжения, кривая угла крена стала элегантнее и гладкой и имел меньше колебания.

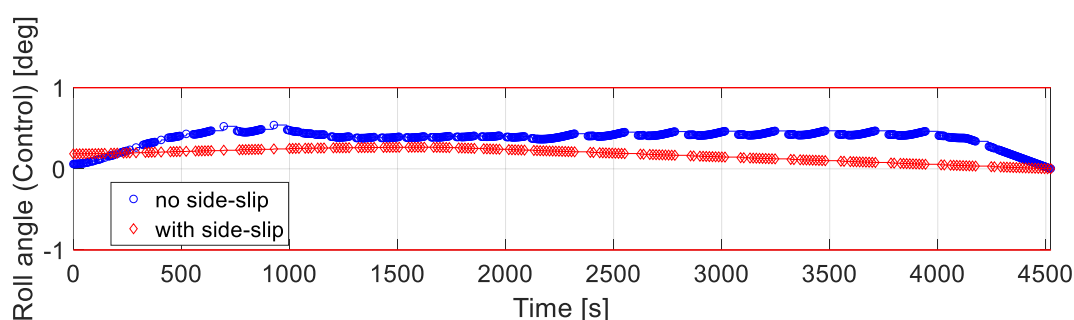


Рисунок 4.13 угол крена при и без наличия действие угла скольжения

Угол атаки не изменился на траектории кроме точек начала и конца. Сравнения угла атаки поставили в рисунке 4.14. Закон угла скольжения показан в рисунке 4.15, что он имел мелкие значения в порядке 0.01 градуса.

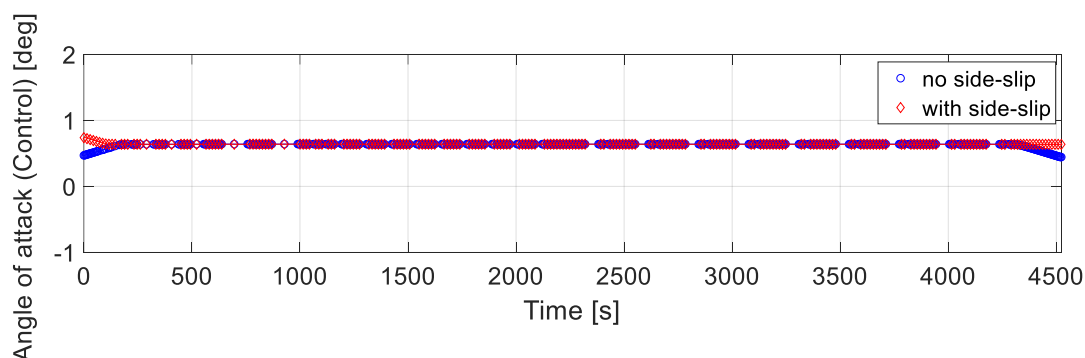


Рисунок 4.14 угол атака при и без наличия действие угла скольжения

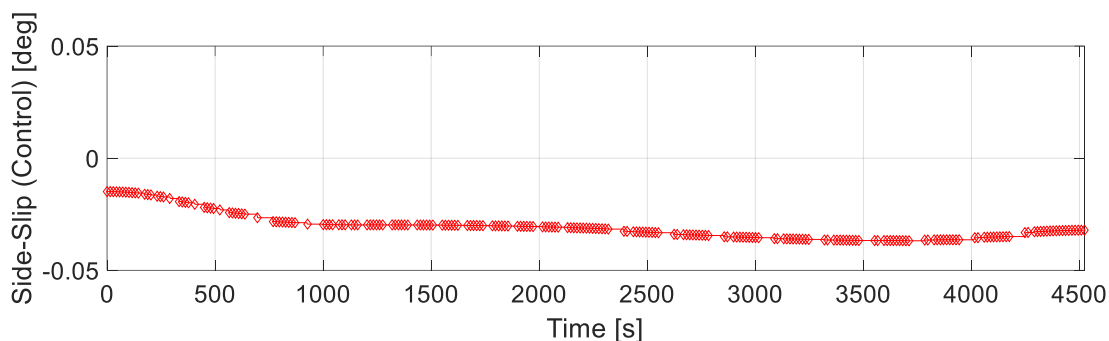


Рисунок 4.15 закон угол скольжения – первая траектория

4.2.2 Результаты второй траекторий ВГ

Изменявшиеся параметры являются угол крена, угол атака и угол скольжения сам. В рисунке 4.16 показали сравнение угол крена траектории при наличии действия угла скольжения (красная кривая) и без наличия действия угла скольжения (синяя кривая). Заметим, что разница между обеим кривыми угла крена не большая.

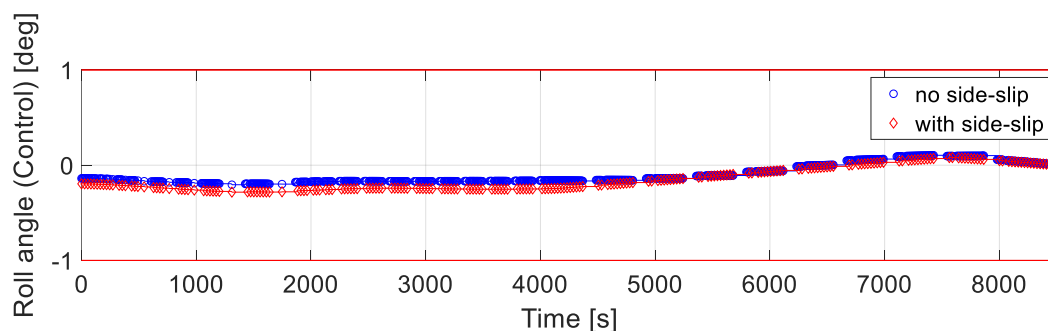


Рисунок 4.16 угол крена при и без наличия действие угла скольжения

Угол атака тоже не изменился на траектории кроме точек начала и конца. Сравнения угла атака поставили в рисунке 4.17. Закон угла скольжения показан в рисунке 4.18, что он имел тоже мелкие значения в порядке 0.05 градуса.

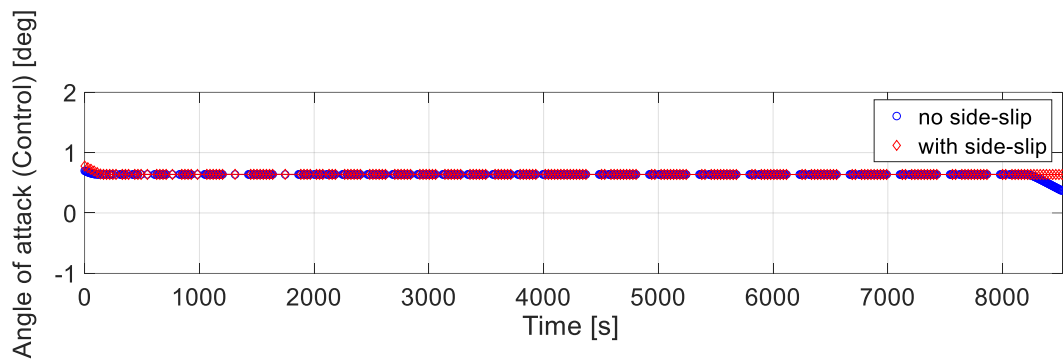


Рисунок 4.17 угол атака при и без наличия действие угла скольжения

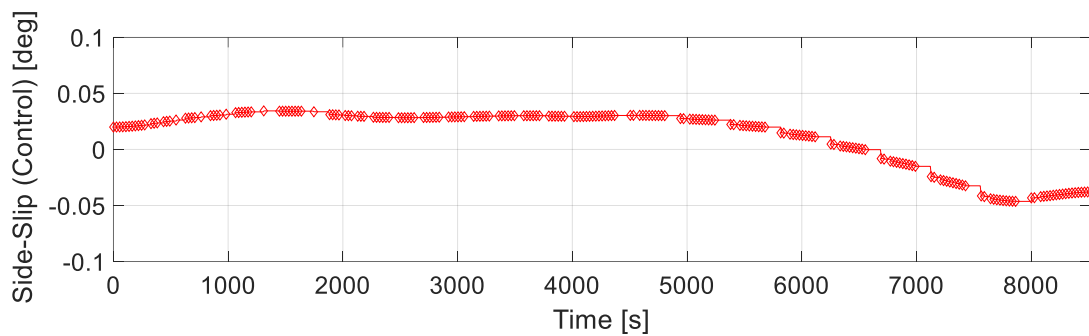


Рисунок 4.18 закон угол скольжения – вторая траектория

Из прошедших результатов заметим, что значения угла скольжения очень мелкие, поэтому его действие на других параметров не большое и его действие на траектории не ощущено. На базе этого сможем сказать, что добывание угла скольжения в качестве параметр управления только усложняет задачу и не добывает ту замечательную точность.

Глава 5 Реализация программы управления в АБСУ

5.1 Система АБСУ:

Автоматическая бортовая система управления (АБСУ) обеспечивается [28,29] заданные характеристики устойчивости и управляемости самолета на всех режимах полета; Автоматизации управления самолетом на всех этапах полета (набора высоты, снижения, а также маршрутного полета); Обеспечения автоматического и детекторного управления самолетом при заходе на посадку.

АБСУ функционально состоит из системы штурвального управления (СШУ), системы автоматического управления (САУ), системы автоматического и директорного захода на посадку (система траекторного управления – СТУ), системы встроенного контроля (СВК), автомата тяги (АТ).

Система штурвального управления обеспечивает требуемые характеристики устойчивости и управляемости самолёта в штурвальном режиме путём дополнительного перемещения рулей самолёта и дополнительной загрузки органов управления. Состоит из системы устойчивости и управляемости (СУУ) и системы автоматической балансировки (САБ).

5.2 Входящие и выходящие параметры программы управления

Программ управления имеет следующие начальные входы, см. приложение В.2, высота, скорость полета, масса топлива, углы широты и долготы, углы атаки, скольжения и крена, и положение дроссельного клапана. В течении полёта требуется определить значению плотности и величину земного ускорения. При фиксированной высоты последние два параметра получит одно величине.

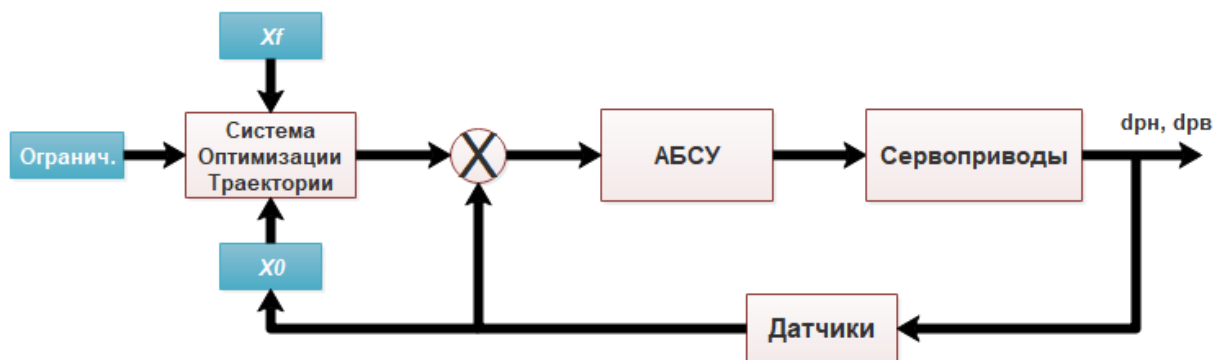
Целевое место определяем по трем параметрам (высота, углы широты и долготы). Эти три параметра имеют разные значения на оптимальной траектории, чтобы обеспечивают минимизации конечной массы топлива.

Выходящие параметры кода - углы атаки, скольжения и крена, положения дроссельного клапана, углы наклона траектории, скорость, расход топлива. Самолёт должен получить эти выходящие параметры, чтобы назвать её траекторию оптимальным т.е. чтобы минимизировать конечный расход топлива.

5.3 Реализация в АБСУ

В диаграмме 3.1, Система оптимизации траектории имеет три входа (x_0 – начальные значения, x_f – конечные значения и ограничения траектории) и один векторный выход содержит: высота, скорость, углы наклона траектории, углы атаки, скольжения и крена, и положение дроссельного клапана. Нам интересуют выход сервоприводов $\delta_{рн}$ – наклона руля направления и $\delta_{рв}$ – наклона руля высоты. От датчиков сформируем обратную связь.

Диаграмма 3.1. Реализация программы управления а АБСУ



После ввода требования система оптимизации траектории (COT), эта система сразу решит задачу и даст рекомендации для получения оптимальной траектории, эти рекомендации по форме дискретных величин переменных движения. Автоматизированная бортовая система управления должна обеспечивать этих переменных через компьютеры АБСУ, которые дают сигналы управления к поверхностям самолёта через элементы действия (сервоприводы).

Внутри компьютера АБСУ должно существовать контролер, который её работа является регулированием положения поверхности самолёта с значением данных переменных от СОТ. Закон отклонения поверхностей самолёта в рыскании и тангаже может иметь следующие законы (5.1).

$$\delta = \delta_0 + \Delta\delta_{\text{лет}} + \Delta\delta_{\text{АБСУ.норм}} + \Delta\delta_{\text{АБСУ.опт}} \quad (5.1)$$

Где δ_0 начальное положение назначенной поверхности самолёта, $\Delta\delta_{\text{лет}}$ отклонение назначенной поверхности от лётчика, $\Delta\delta_{\text{АБСУ.норм}}$ обычное (без оптимизации) отклонение назначенной поверхности от АБСУ, $\Delta\delta_{\text{АБСУ.опт}}$ оптимальное отклонение назначенной поверхности от АБСУ для обеспечения оптимальной траектории.

ЗАКЛЮЧЕНИЕ

Проведённые исследования программ и алгоритмов управления движением гиперзвукового летательного аппарата на участке горизонтального крейсерского полета и полученные результаты позволяют сделать следующие основные выводы.

Метод НЛП с помощью алгоритма внутренней точки (МВТ) и с использованием программа трех переменных управления (угол атаки, угол крена и положение дроссельного клапана) позволяет минимизировать расходы топлива гиперзвукового летательного аппарата при выполнении заданных конечных условий движения по высоте, скорости, углы наклона траектории и углы широты и долготы.

Пользование программа четырёх переменных управления (угол атаки, угол скольжения, угол крена и положение дроссельного клапана) позволяет также минимизировать расходы топлива гиперзвукового летательного аппарата при выполнении заданных конечных условий движения. Четыре переменных управления дают лучшие результаты, но разница не большая сравнительно с программой трех переменных управления, и пользоваться такую программу только усложняет задачу.

Метод максимума Понтрягина является сложной задачей и требует длинные символьные расчёты и добивание коды для регулирования условиях Понтрягина. В этом методе также требуется решить дифракционные уравнения множителей Лагранжа, и определить их начальные значение, эти множители не имеют не какие физические значения и сложно гадать их начальные значения.

Поскольку предложенные программы и алгоритмы управления инвариантны к заданным массовым, геометрическим и аэродинамическим характеристикам, характеристикам силовой установки, конечным условиям движения, ограничениям на управление и параметры движения, то они имеют

универсальный характер и могут быть использованы в динамическом проектировании гиперзвуковых летательных аппаратов различного назначения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кочян А. Г. Программы и алгоритмы управления движением гиперзвукового летательного аппарата на участке разгона – набора высоты // Студенческая наука аэрокосмическому комплексу. Самара, Самарский государственный аэрокосмический университет. – Диссертация, 2009. – 210 с.
2. Цихош Э. Сверхзвуковые самолёты // Москва «Мир» / перевод Левицкого Л. В., под редакцией Микладзе В. Г. – Книга. 1983. – 425 с.
3. Ramadi B. N. Performance, stability, dynamics and control of airplanes – second edition // American institute of aeronautics and astronautics / NASA Langley research center, Hampton, Virginia. Под редакцией Schetz J. A. – Книга. 2014. – 801 с.
4. Cai G., Chen B. M., Lee T. H. Unmanned rotorcraft systems // Springer publications / Глава 2, Coordinate Systems and Transformations. – Книга. 2011. – 270 с.
5. Zipfel P. H. Modeling and simulation of aerospace vehicle dynamics – second edition // American institute of aeronautics and astronautics / University of Florida. Под редакцией Schetz J. A. – Книга. 2007. – 607 с.
6. Weiland C. Computational space flight mechanics // Springer publications: Verlag Berlin Heidelberg. – Книга. 2010. – 315 с.
7. Nancy Hall (ed.). "Mach Number". NASA, <https://www.grc.nasa.gov/www/k-12/airplane/mach.html>. Дата входа 29.10.2018.
8. Кулазин В. В. Теория, расчет и проектирование авиационных двигателей и энергетических установок – первая книга // Москва: машиностроение. – Книга. 2003. – 618 с.
9. Справочник атмосферы – справочники данные модели // под ред. Ю. С. Седунова. – Ленинград: Гидрометеиздат, – Справочник, 1991. – 510 с.
10. Летов А. М. Динамика полёта и управление. – М.: Наука, 1969. – 360 с.
11. Murillo O. J. A fast ascent trajectory optimization method for hypersonic air-breathing vehicles // Iowa state university: Aerospace Engineering. – Диссертация, 2010. – 154 с.

12. Lu P., Sun H., Tsai B. Closed-Loop Endo-Atmospheric Ascent Guidance // Journal of Guidance Control and Dynamics. – Статья, 2003. – 33 с.
13. Pontryagin L. S., Boltyanskii V. G., Gamkrelidze R. V., and Mishchenko E. F. The Mathematical Theory of Optimal Processes // John Willy, translated from Russian. – Книга, 1962. – 362 с.
14. https://www.encyclopediaofmath.org/index.php/Lagrange_problem, Задача Лагранжа / Математическая энциклопедия. Дата входа 7.11.2018.
15. Connor J. Ю., Naughton A. P., Symbolic Computation - “All you will ever need to know about Maple”. – Книга, 2009. – 129 с.
16. Shampine L. F., Gladwell I., Thompson S. Solving ODEs with MATLAB. – Книга, 2003. – 273 с.
17. Janicki W. D. Asymptotic analysis of hypersonic vehicle dynamics along entry trajectory // U.S. Air Force Academy: Astronautical Engineering., - Диссертация, 1989. – 85 с.
18. Betts J. T. Practical methods for optimal control and estimation using nonlinear programming // North-Carolina state university. – Под ред. Smith R. C. – Книга, 2010. – 449 с.
19. Speyer J.L., Jacobson D.H, Primer on optimal control theory // North-Carolina state university. – Под ред. Smith R. C. – Книга, 2010. – 314 с.
20. Donald E. K. Optimal control theory - an introduction // State university of San-Jose California. - Книга, 2004. – 472 с.
21. Enrico B. Numerical optimal control // Department of industrial engineering, – University of Trento. – Книга, 2015. – 429 с.
22. Longuski J. M. Optimal control theory with aerospace applications // Springer Publications. – Книга, 2014. С. 19 – 38.
23. Tadeusz J. M. Multi objective trajectory optimization of a hypersonic reconnaissance vehicle GHAME with temperature constraints // AIR University: Air force institute of technology. – Диссертация, 2014. 416 с.
24. Bryson A. E., HO Y. C., Applied optimal control // John-Wiley and Sons: New York. – Книга, 1975. 329 с.

25. Zhang X. Matrix analysis and applications // Cambridge University press. – Книга, 2017.
26. Nocedal J., Wachter A., Waltz R. Adaptive barrier update strategies for nonlinear interior methods // SIAM Journal on Optimization. – Статья. 2006. 20 с.
27. Vanderbei R. J. Interior point methods and nonlinear optimization // <http://www.princeton.edu/~rvdb>, дата входа 12.12.2018. – Лекции. 2012. 51с.
28. Коптев А.Н. Авиационное и радиоэлектронное оборудование воздушных судов гражданской авиации, книга 3 // Самара издательство СГАУ. – Учебное пособие. 2011. 392с.
29. Boeing B747-400 Auto flight systems // ATA 22 EASA part-66 B1. – Справочник. 218с.
30. СТО СГАУ 02068410-004-2007. Общие требования к учебным текстовым документам.– Введ. 2007. –10. –09. – Самара: СГАУ, 2007. – 34с.

Приложение А: Гиперзвуковые Силовые Установки

ГЛА летает на скорости трёх диапазонов Маха (дозвуковой $< 0,8$, трансзвуковой $0,8 - 1,3$, сверхзвуковой $1,3 - 5$) чтобы достичь гиперзвуковой скорости из диапазона $5 - 25$, [7], поэтому необходим двигатель, который сможет работать на этих четырёх диапазонах числа Маха, но из-за отсутствия таких двигателей, рассмотрим разные двигатели, работающие в каждом диапазоне.

Основные типы двигатели и комбинации между ними представлена на рисунке А.1, [1,8].

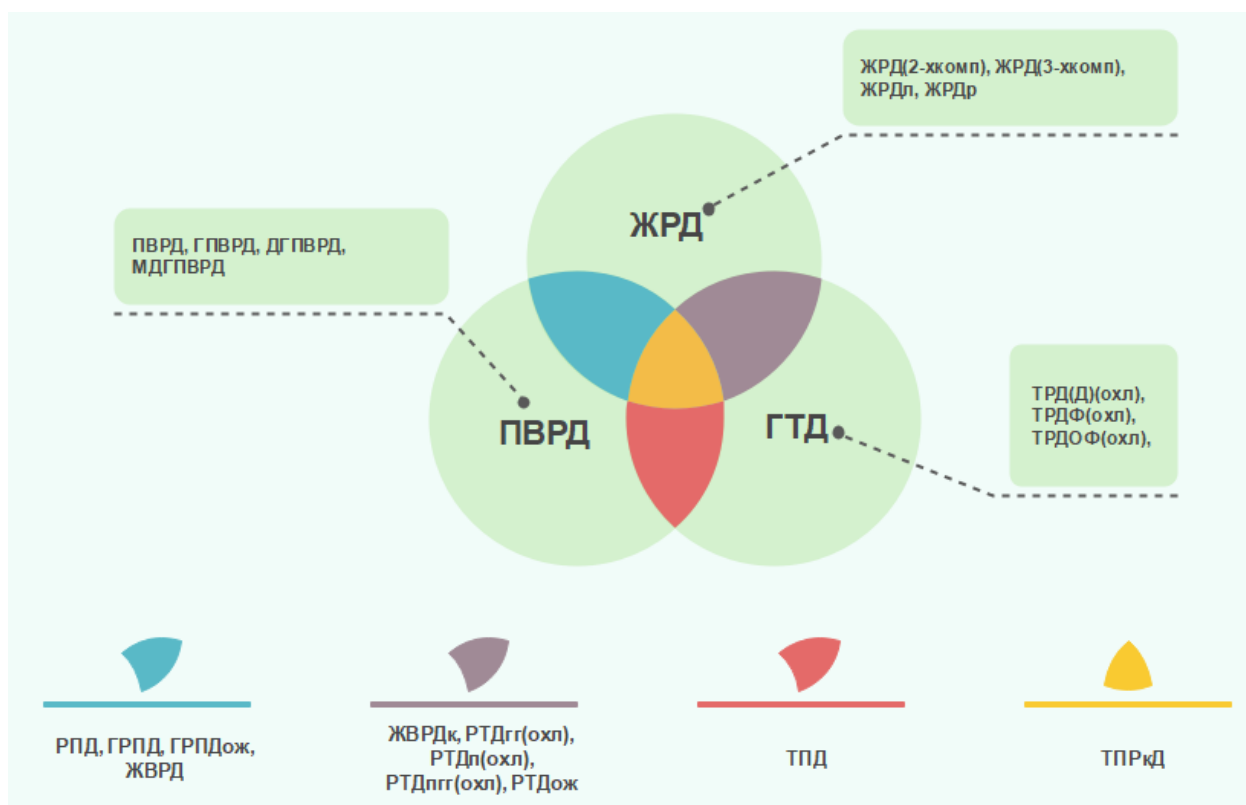


Рисунок А.1 Основные типы двигателей и комбинации между ними

Где **ПВРД** - прямоточный воздушно-реактивный двигатель. **ГПВРД** - гиперзвуковой ПВРД. **ДГПВРД** - двухрежимный ГПВРД. **МГПВРД** - многорежимный ГПВРД. **ГТД** - газотурбинный двигатель. **ТРДД** - двухконтурный турбореактивный двигатель. **ТРДФ** - турбореактивный двигатель с форсажной камерой. **ТРДДФ** - ТРДД с форсажной камерой. **ТРДОФ** - ТРДФ с отбором воздуха за компрессором. **РТД** - ракетно-турбинный двигатель

(«гг» - газогенераторной схемы, «п» - пароводородной схемы, «пгг» - парогенераторной схемы, «ож» - с ожижением части расхода воздуха). РПД - ракетно-прямоточный двигатель. ГРПД - гиперзвуковой РПД (ДГРПД - двухрежимный ГРПД, индекс «ож» - с ожижением части расхода воздуха). ЖРД - жидкостно-ракетный двигатель (2-хкомп - двухкомпонентный, 3-хкомп - трёхкомпонентный, индекс «л» - линейной схемы, «р» - с раздвижным соплом).

В рисунке А.2 поставили зависимости удельного импульса от числа Маха, где при знании удельного импульса и расход топлива сможем писать приблизительный модель ГПТД, (см. глава 1 часть 3 – моделирования СУ).

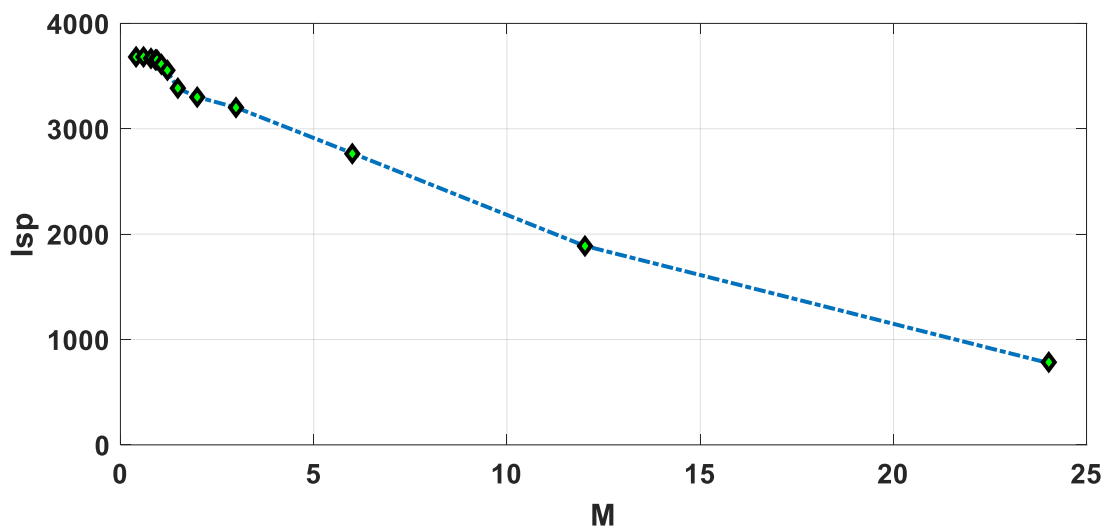


Рисунок А.2 зависимости удельного импульса I_{ui} от числа Маха

Приложение Б: расширенные уравнения

Уравнение функции Лагранжа:

$$L(x, v, \lambda, s) = -m_{\text{кон. сам.}} - [v_1 \ v_2 \ v_3 \ v_4 \ v_5 \ v_6 \ v_7]^T * [$$

$$V_{k+1} - V_k - h * \left(\frac{T_x - D}{m} - g \sin(\gamma) + V_{om} \right);$$

$$\gamma_{k+1} - \gamma_k - \frac{h}{V} \left(\frac{L \cos(\mu_a) - Y_a \sin(\mu_a)}{m} - g \cos(\gamma) + \gamma_{om} \right);$$

$$\chi_{k+1} - \chi_k - \frac{h}{V} \left(\frac{L \sin(\mu_a) + Y_a \cos(\mu_a)}{m \cos(\gamma)} + \chi_{om} \right);$$

$$h_{k+1} - h_k - h V \sin(\gamma);$$

$$\theta_{k+1} - \theta_k - h \frac{V \cos(\gamma) \sin(\chi)}{(R \cos(\phi))};$$

$$\phi_{k+1} - \phi_k - h \frac{V \cos(\gamma) \sin(\chi)}{R};$$

$$m_{k+1} - m_k - h (v_{\text{ном}} + (v_{\text{макс}} - v_{\text{ном}}) * \eta_t)] -$$

$$-[\lambda_1 \ \lambda_2 \ \lambda_3 \ \lambda_4 \ \lambda_5 \ \lambda_6 \ \lambda_7 \ \lambda_8 \ \lambda_9 \ \lambda_{10} \ \lambda_{11} \ \lambda_{12} \ \lambda_{13} \ \lambda_{14} \ \lambda_{15} \ \lambda_{16} \ \lambda_{17} \ \lambda_{18} \ \lambda_{19} \ \lambda_{20}]^T *$$

$$[+R - 30000 - s_1; -R + 30000 - s_2; +\gamma - 1.55 - s_3 = 0; -\gamma + 1.55 - s_4 = 0;$$

$$+\chi - 3.14 - s_5 = 0; -\chi + 3.14 - s_6 = 0; +\mu_a - 0.087 - s_7 = 0; -\mu_a + 0.087 - s_8 = 0;$$

$$+V - 5 - s_9 = 0; -V + 8 - s_{10} = 0; +m - 54,431.07 - s_{11} = 0;$$

$$-m + 136,077.7 - s_{12} = 0; +\alpha - 0.17 - s_{13} = 0; -\alpha + 0.17 - s_{14} = 0;$$

$$+\beta - 0.17 - s_{15} = 0; -\beta + 0.17 - s_{16} = 0; +\phi - A_\phi - s_{17} = 0;$$

$$-\phi + B_\phi - s_{18} = 0; +\theta - A_\theta - s_{19} = 0; -\theta - B_\theta - s_{20} = 0] -$$

$$-\mu \sum_{i=1}^{20} \log(s_i) \quad (\text{C. 1})$$

Приложение В: коды МАТЛАБ

Программ решение задачи оптимизации состоит из разных код, которые написали в МАТЛАБ, и положили в этом приложении. Над каждой строкой написали описание строку.

В.1 Основной код

```
% Main script to solve the Optimal Control Problem
% Hypersonic Aircraft Minimum Fuel Cruising
%-----

clc; clear all; close all; format compact;
global sol;
sol=[]; % Initialize solution
structure

options = settings(40); % Get options and solver
settings
[problem,guess] = MinFuelFlight; % Fetch the problem
definition
errorHistory = zeros(2,length(problem.states.x0));
npsegmentHistory = zeros(2,1);
ConstraintErrorHistory =
zeros(2,length(problem.constraintErrorTol));
timeHistory = zeros(1,2);
iterHistory = zeros(1,2);

solutionHistory = cell(1,2);

maxAbsError = 1e9;
i=1; imax=4; % 50 recommended

problem_iter=problem;
while (any(maxAbsError>problem.states.xErrorTol) ||
any(maxAbsConstraintError>problem.constraintErrorTol)) && i<=imax

[infoNLP,data,options]=transcribeOCP(problem_iter,guess,options);
% Format for NLP solver
[solution,status,data] = solveNLP(infoNLP,data);
% Solve the NLP
[solution]=output(problem,solution,options,data,4);
% Output solutions

maxAbsError=max(abs(solution.Error));
maxAbsConstraintError=max(solution.ConstraintError);
errorHistory(i,:)=maxAbsError;
iterHistory(i)=status.iter;
ConstraintErrorHistory(i,:)=maxAbsConstraintError;
```

```

timeHistory(i)=solution.computation_time;
solutionHistory{i}=solution;

    if (any(maxAbsError>problem.states.xErrorTol) ||
any(maxAbsConstraintError>problem.constraintErrorTol)) && i<=imax
        [ options, guess ] = doMeshRefinement( options, problem,
guess, data, solution, i );
    end
    i=i+1;
end

MeshRefinementHistory.errorHistory=errorHistory;
MeshRefinementHistory.timeHistory=timeHistory;
MeshRefinementHistory.ConstraintErrorHistory=ConstraintErrorHistory;

%%% Forming Plots
xx=linspace(solution.T(1,1),solution.tf,1000);

figure
subplot(211);plot([solution.T(:,1);solution.tf],speval(solution.Xp,2,[solution.T(:,1);solution.tf])*180/pi,'bo-'); grid on
xlabel('Time'); ylabel('Latitude [deg]')
subplot(212);plot([solution.T(:,1);solution.tf],speval(solution.Xp,3,[solution.T(:,1);solution.tf])*180/pi,'go-'); grid on
xlabel('Time'); ylabel('Longitude [deg]')

figure
plot(speval(solution.Xp,3,[solution.T(:,1);solution.tf])*180/pi,speval(solution.Xp,2,[solution.T(:,1);solution.tf])*180/pi,'b-','LineWidth',2)
xlabel('East Position [m]')
ylabel('North Position [m]')
grid on
%%% Starting Point and Final Point
hold on
plot(problem.states.x0(3)*180/pi, problem.states.x0(2)*180/pi,'.k', 'MarkerSize',20)
plot(problem.states.xf1(3)*180/pi, problem.states.xf1(2)*180/pi,'.k', 'MarkerSize',20)
text(problem.states.x0(3)*180/pi,problem.states.x0(2)*180/pi,'ORG')
text(problem.states.xf1(3)*180/pi,problem.states.xf1(2)*180/pi,'DES')
xlim([-180 180])
ylim([-90 90])
hold off

figure
% Altitude
subplot(2,1,1)

```

```

plot([solution.T(:,1);
solution.tf],speval(solution.Xp,1,[solution.T(:,1);
solution.tf]), 'bo' )
hold on
plot(xx,speval(solution.Xp,1,xx), 'b-' )
% plot([solution.T(1,1); solution.tf],[problem.states.xl(1),
problem.states.xl(1)], 'r-' )
plot([solution.T(1,1); solution.tf],[problem.states.xu(1),
problem.states.xu(1)], 'r-' )
xlim([0 solution.tf])
xlabel('Time [s]')
ylabel('Altitude [m]')
grid on

subplot(2,1,2)
% True Airspeed
plot([solution.T(:,1);
solution.tf],speval(solution.Xp,4,[solution.T(:,1);
solution.tf]), 'bo' )
hold on
plot(xx,speval(solution.Xp,4,xx), 'b-' )
plot([solution.T(1,1); solution.tf],[problem.states.xl(4),
problem.states.xl(4)], 'r-' )
plot([solution.T(1,1); solution.tf],[problem.states.xu(4),
problem.states.xu(4)], 'r-' )
xlim([0 solution.tf])
xlabel('Time [s]')
ylabel('True Airspeed [m/s]')
grid on

figure
subplot(2,1,1)
% Flight Path Angle
plot([solution.T(:,1);
solution.tf],speval(solution.Xp,5,[solution.T(:,1);
solution.tf])*180/pi, 'bo' )
hold on
plot(xx,speval(solution.Xp,5,xx)*180/pi, 'b-' )
hold on
xlim([0 solution.tf])
xlabel('Time [s]')
ylabel('Flight Path Angle [deg]')
grid on

subplot(2,1,2)
% Tracking Angle
plot([solution.T(:,1);
solution.tf],speval(solution.Xp,6,[solution.T(:,1);
solution.tf])*180/pi, 'bo' )
hold on
plot(xx,speval(solution.Xp,6,xx)*180/pi, 'b-' )
hold on

```

```

xlim([0 solution.tf])
xlabel('Time [s]')
ylabel('Tracking Angle [deg]')
grid on

figure
subplot(2,1,1)
% Aircraft Mass
plot([solution.T(:,1);
solution.tf],speval(solution.Xp,7,[solution.T(:,1);
solution.tf]),'bo' )
hold on
plot(xx,speval(solution.Xp,7,xx),'b-' )
xlim([0 solution.tf])
xlabel('Time [s]')
ylabel('Aircraft Mass [kg]')
grid on

subplot(2,1,2)
% Angle of attack (Control)
plot(solution.T(:,1),speval(solution.Up,1,solution.T)*180/pi,'bo'
)
hold on
plot(xx,speval(solution.Up,1,xx)*180/pi,'b-' )
plot([solution.T(1,1); solution.tf],[problem.inputs.ul(1),
problem.inputs.ul(1)]*180/pi,'r-' )
plot([solution.T(1,1); solution.tf],[problem.inputs.uu(1),
problem.inputs.uu(1)]*180/pi,'r-' )
xlim([0 solution.tf])
xlabel('Time [s]')
ylabel('Angle of attack (Control) [deg]')
grid on

figure
subplot(2,1,1)
% Roll angle (Control)
plot(solution.T(:,1),speval(solution.Up,2,solution.T)*180/pi,'bo'
)
hold on
plot(xx,speval(solution.Up,2,xx)*180/pi,'b-' )
plot([solution.T(1,1); solution.tf],[problem.inputs.ul(2),
problem.inputs.ul(2)]*180/pi,'r-' )
plot([solution.T(1,1); solution.tf],[problem.inputs.uu(2),
problem.inputs.uu(2)]*180/pi,'r-' )
xlim([0 solution.tf])
xlabel('Time [s]')
ylabel('Roll angle (Control) [deg]')
grid on

subplot(2,1,2)
% Throttle Setting (Control)
plot(solution.T(:,1),speval(solution.Up,3,solution.T),'bo')

```



```

hold on
plot(xx,speval(solution.Up,3,xx),'b-' )
plot([solution.T(1,1); solution.tf],[problem.inputs.ul(3),
problem.inputs.ul(3)],'r-' )
plot([solution.T(1,1); solution.tf],[problem.inputs.uu(3),
problem.inputs.uu(3)],'r-' )
xlim([0 solution.tf])
xlabel('Time [s]')
ylabel('Throttle Setting (Control) [-]')
grid on

```

B.2 Код уравнений и ограничений

```

function [problem,guess] = MinFuelFlight
% Minimum Fuel Flight Profile
%----- BEGIN CODE -----
% Plant model name
problem.data.plantmodel = 'MinFuelFlight';

% Constants
auxdata.g=9.81;
auxdata.omegae=7.27199e-5 ;
auxdata.rearth=6.38E6;

%--Initial Conditions--          --Terminal Conditions--
W0=136077.7;

H0=30000;                        Hf=30000;
Vtas0=1525;                     Vtasf=1830;

%%% Profile A
% PPH0=55*pi/180;                PPHf=41*pi/180;
% PTH0=38*pi/180;                PTHf=-74*pi/180;

%%% Profile B
PPH0=-13.05*pi/180;             PPHf=-18.19*pi/180;
PTH0=-45.25*pi/180;            PTHf=130.54*pi/180;
chi0=0*pi/180;                 chif=-3/4*pi/180;
gamma0=2*pi/180;               gammaf=-3*pi/180;

alpha0=2*pi/180;
phi_0 = 0.5*pi/180;
throttle_0 = 0.5;

% variable bounds
% Put PPH_max & min and PTH_max & min equals to PPH0&f and PTH0&f
%-----MIN-----              -----MAX-----
W_min=(136077.7-81646.63);

H_min=H0;                       H_max=30000;
Vtas_min=Vtas0;                 Vtas_max=1830;
gamma_min=-30*pi/180;           gamma_max=30*pi/180;
chi_min=-180*pi/180;            chi_max=180*pi/180;

```

```

%%% Profile A
% PPH_min=min(PPH0,PPHf);          PPH_max=max(PPH0,PPHf);
% PTH_min=min(PTH0,PTHf);          PTH_max=max(PTH0,PTHf);
%%% Profile B
PPH_min=min(PPH0,PPHf);            PPH_max=max(PPH0,PPHf);
PTH_min=min(PTH0,PTHf);            PTH_max=max(PTH0,PTHf);
alpha_min=-10*pi/180;              alpha_max=10*pi/180;
phi_min=-1*pi/180;                 phi_max=1*pi/180;
throttle_min=0;                    throttle_max=1;

% Parameter for the obstacle
auxdata.obs_npos_1=0;auxdata.obs_epos_1=0;auxdata.obs_r_1=0;
auxdata.obs_npos_2=0;auxdata.obs_epos_2=0;auxdata.obs_r_2=0;
auxdata.obs_npos_3=0;auxdata.obs_epos_3=0;auxdata.obs_r_3=0;
auxdata.obs_npos_4=0;auxdata.obs_epos_4=0;auxdata.obs_r_4=0;
auxdata.obs_npos_5=0;auxdata.obs_epos_5=0;auxdata.obs_r_5=0;

%% Time
% Initial Time. t0 < tf
% Final time. Let tf_min=tf_max if tf is fixed
problem.time.t0=0; % Initial t
%%% Profile A
% guess.tf=4523; % Guess 4523
% problem.time.tf_min=4523; problem.time.tf_max=4523;
%%% Profile B
guess.tf=8519; % Guess 8519
problem.time.tf_min=8519; problem.time.tf_max=8519;

% Parameters bounds. pl=< p <=pu
problem.parameters.pl=[];
problem.parameters.pu=[];
guess.parameters=[];

% Initial conditions for system.
problem.states.x0=[H0 PPH0 PTH0 Vtas0 gamma0 chi0 W0];

% Initial conditions for system. Bounds if x0 is free s.t. x0l=<
x0 <=x0u
problem.states.x0l=[H0 PPH0 PTH0 Vtas0 gamma0 chi0 W0];
problem.states.x0u=[H0 PPH0 PTH0 Vtas0 gamma0 chi0 W0];

% State bounds. xl=< x <=xu
problem.states.xl=[H_min PPH_min PTH_min Vtas_min gamma_min
chi_min W_min];
problem.states.xu=[H_max PPH_max PTH_max Vtas_max gamma_max
chi_max W0];

% State rate bounds. xrl=< x <=xru
problem.states.xrl=[-inf -deg2rad(0.0001) -deg2rad(0.0001) -inf -
deg2rad(0.01) -deg2rad(0.01) -inf];
problem.states.xru=[inf deg2rad(0.0001) deg2rad(0.0001) inf
deg2rad(0.01) deg2rad(0.01) inf];

```

```

% State error bounds
problem.states.xErrorTol=[5 deg2rad(0.0001) deg2rad(0.0001) 0.5
deg2rad(0.01) deg2rad(0.01) 0.1*auxdata.g];

% State constraint error bounds
problem.states.xConstraintTol=[5 deg2rad(0.0001) deg2rad(0.0001)
0.5 deg2rad(0.01) deg2rad(0.01) 0.1*auxdata.g];
problem.states.xrConstraintTol=[5 deg2rad(0.0001) deg2rad(0.0001)
0.5 deg2rad(0.01) deg2rad(0.01) 0.1*auxdata.g];

% Terminal state bounds. xfl=< xf <=xfu
problem.states.xfl=[Hf PPHf PTHf Vtasf gammaf chif W_min];
problem.states.xfu=[Hf PPHf PTHf Vtasf gammaf chif W0];

% Guess the state trajectories with [x0 xf]
guess.time=[0 guess.tf/2 guess.tf];
guess.states(:,1)=[H_max H_max/2 H_max];
guess.states(:,2)=[PPH0 PPHf/2 PPHf];
guess.states(:,3)=[PTH0 PTHf/2 PTHf];
guess.states(:,4)=[Vtas_max Vtas_max/2 Vtas_max];
guess.states(:,5)=[gamma0 gammaf/2 gammaf];
guess.states(:,6)=[chi0 chif/2 chif];
%% Profile A and B
guess.states(:,7)=[W0 (136077.7-81646.63)/2 (136077.7-81646.63)];

% Number of control actions N
problem.inputs.N=0;

% Input bounds
problem.inputs.ul=[alpha_min phi_min throttle_min];
problem.inputs.uu=[alpha_max phi_max throttle_max];

problem.inputs.u0l=[alpha_min phi_min throttle_min];
problem.inputs.u0u=[alpha_max phi_max throttle_max];

% Input rate bounds
problem.inputs.url=[-deg2rad(2) -deg2rad(10) -inf];
problem.inputs.uru=[deg2rad(2) deg2rad(10) inf];

% Input constraint error bounds
problem.inputs.uConstraintTol=[deg2rad(0.5) deg2rad(0.5) 0.1];
problem.inputs.urConstraintTol=[deg2rad(0.5) deg2rad(0.5) 0.1];

% Guess the input sequences with [u0 uf]
guess.inputs(:,1)=[alpha0 alpha0 alpha0];
guess.inputs(:,2)=[phi_0 phi_0 phi_0];
guess.inputs(:,3)=[throttle_0 throttle_0 throttle_0];

% Choose the set-points if required
problem.setpoints.states=[];
problem.setpoints.inputs=[];

```

```

% Bounds for path constraint function  $g_l \leq g(x,u,p,t) \leq g_u$ 
problem.constraints.gl=[0 0 0 0 0];
problem.constraints.gu=[inf inf inf inf inf];
problem.constraints.gTol=[(auxdata.obs_r_1+1)^2-
(auxdata.obs_r_1)^2 (auxdata.obs_r_2+1)^2-(auxdata.obs_r_2)^2
(auxdata.obs_r_3+1)^2-(auxdata.obs_r_3)^2 (auxdata.obs_r_4+1)^2-
(auxdata.obs_r_4)^2 (auxdata.obs_r_5+1)^2-
(auxdata.obs_r_5)^2]/1e04;

% Bounds for boundary constraints  $b_l \leq b(x_0,x_f,u_0,u_f,p,t_0,t_f) \leq b_u$ 
problem.constraints.bl=[0];
problem.constraints.bu=[0];

% store the necessary problem parameters used in the functions
problem.data.auxdata=auxdata;

% Get function handles and return to Main.m
problem.functions={@L,@E,@f,@g,@avrc,@b};
problem.functions_unscaled={@L_unscaled,@E_unscaled,@f_unscaled,@
g_unscaled,@avrc_unscaled,@b_unscaled};
problem.constraintErrorTol=[problem.constraints.gTol,problem.constraints.gTol,problem.states.xConstraintTol,problem.states.xConstraintTol,problem.inputs.uConstraintTol,problem.inputs.uConstraintTol];
%----- END OF CODE -----

%      x - state vector
%      xr - state reference
%      u - input
%      ur - input reference
%      p - parameter
%      t - time
%      x0 - state at t=0
%      xf - state at t=tf
%      u0 - input at t=0
%      uf - input at t=tf
%      p - parameter
%      tf - final time
%      data- structured variable containing the values of
additional data used inside the function

function stageCost=L_unscaled(x,xr,u,ur,p,t,data)
% L - Returns the stage cost. The function is vectorized and
% xi, ui are column vectors taken as x(:,i) and u(:,i) (i denotes
the i-th
% variable)
%
% Output: stageCost - Scalar or vectorized stage cost
%----- BEGIN CODE -----
stageCost = 0*t;

```

```

%----- END OF CODE -----

function boundaryCost=E_unscaled(x0,xf,u0,uf,p,t0,tf,data)
% E - Returns the boundary value cost
% Syntax:  boundaryCost=E(x0,xf,u0,uf,p,tf,data)
% Output:  boundaryCost - Scalar boundary cost
%----- BEGIN CODE -----
boundaryCost= -xf(7);
%----- END OF CODE -----

function dx = f_unscaled(x,u,p,t,data)
% f - Returns the ODE right hand side where x'= f(x,u,p,t)
% The function must be vectorized and xi, ui, pi are column
vectors taken as x(:,i), u(:,i) and p(:,i).
% Each state corresponds to one column of dx.
%
% Syntax:  dx = f(x,u,p,t,data)
% Output:  dx - time derivative of x
%----- BEGIN CODE -----
auxdata = data.auxdata;

H = x(:,1);
PPH = x(:,2);
PTH = x(:,3);
V = x(:,4);
gamma = x(:,5);
chi = x(:,6);
m = x(:,7);
alpha = u(:,1);
phi = u(:,2);
throttle = u(:,3);
ct = 0.05;
cw = 1.00;
% cp = 1; %%% Profile A
cp = 0.09; %%% Profile B
%%% cp = 0.1;%cp = 3.5;%cp = 1;%cp = 1.5;%cp = 0.27;
D = 0.1994.*V.^2.*(-0.0101 + 0.1031.*alpha + 2.1353.*alpha.^2);
% Drag
L = +0.1994.*V.^2.*(-0.0057 + 0.8671.*alpha + 3.5556.*alpha.^2);
% Lift
Thrust = ct*(9.071+(163.29-
9.071).*throttle).*auxdata.g.*(28.8.*(V./305).^3-
0.004315.*V.^2+1.435.*V+3574.2);

%% equations of motions
H_dot      = V.*sin(gamma);
PPH_dot    = cp.*V.*cos(gamma).*sin(chi)./( (auxdata.rearth+H)); %
PPH: Path Phi Angle
PTH_dot    =
V.*cos(gamma).*cos(chi)./( (auxdata.rearth+H).*cos(PPH)); % PTH:
Path Theta Angle

```

```

V_dot      = (Thrust - D - m.*auxdata.g.*sin(gamma))./m +
auxdata.omegae.^2.*(H+auxdata.rearth).*cos(PPH).^2.*(sin(gamma)-
cos(gamma).*tan(PPH).*sin(chi));
gamma_dot  = (1./V).*((L.*cos(phi) -
m.*auxdata.g.*cos(gamma))./(m)+(V.^2.*cos(gamma))./(H+auxdata.rea
rth)+2.*auxdata.omegae.*V.*cos(PPH).*cos(chi)+auxdata.omegae.^2.*
(H+auxdata.rearth).*cos(PPH).^2.*(cos(gamma)+sin(gamma).*tan(PPH)
.*sin(chi)));
chi_dot    = (1./V).*((L.*sin(phi))./(cos(gamma).*m) -
(V.^2./(H+auxdata.rearth)).*cos(gamma).*cos(chi).*tan(PPH)+2.*aux
data.omegae.*V.*(tan(gamma).*cos(PPH).*sin(chi)-sin(PPH))-
(auxdata.omegae.^2.*(H+auxdata.rearth)./cos(gamma)).*sin(PPH).*co
s(PPH).*cos(chi));
W_dot      = -cw*(9.071+(163.29-9.071).*throttle);

dx = [H_dot, PPH_dot, PTH_dot, V_dot, gamma_dot, chi_dot, W_dot];
%----- END OF CODE -----

function c=g_unscaled(x,u,p,t,data)
% g - Returns the path constraint function where gl =< g(x,u,p,t)
=< gu
% The function must be vectorized and xi, ui, pi are column
vectors taken as x(:,i), u(:,i) and p(:,i).
% Each constraint corresponds to one column of c
%
% Syntax:  c=g(x,u,p,t,data)
% Output:  c - constraint function
%----- BEGIN CODE -----

npos = x(:,2);
epos = x(:,3);

c1=(npos-data.auxdata.obs_npos_1).^2+(epos-
data.auxdata.obs_epos_1).^2-data.auxdata.obs_r_1.^2;
c2=(npos-data.auxdata.obs_npos_2).^2+(epos-
data.auxdata.obs_epos_2).^2-data.auxdata.obs_r_2.^2;
c3=(npos-data.auxdata.obs_npos_3).^2+(epos-
data.auxdata.obs_epos_3).^2-data.auxdata.obs_r_3.^2;
c4=(npos-data.auxdata.obs_npos_4).^2+(epos-
data.auxdata.obs_epos_4).^2-data.auxdata.obs_r_4.^2;
c5=(npos-data.auxdata.obs_npos_5).^2+(epos-
data.auxdata.obs_epos_5).^2-data.auxdata.obs_r_5.^2;
c=[c1,c2,c3,c4,c5]/1e04;
%----- END OF CODE -----

function bc=b_unscaled(x0,xf,u0,uf,p,t0,tf,vdat,varargin)
% b - Returns a column vector containing the evaluation of the
boundary constraints: bl =< bf(x0,xf,u0,uf,p,t0,tf) =< bu
%
% Syntax:  bc=b(x0,xf,u0,uf,p,tf,data)

```

```

% Output: bc - column vector containing the evaluation of the
boundary function
%----- BEGIN CODE -----
varargin=varargin{1};
bc=[uf(2)];
%----- END OF CODE -----
% When adaptive time interval add constraint on time
%----- BEGIN CODE -----
if length(varargin)==2
    options=varargin{1};
    t_segment=varargin{2};
    if strcmp(options.transcription,'hpLGR') &&
options.adaptseg==1
        if size(t_segment,1)>size(t_segment,2)
            bc=[bc;diff(t_segment)];
        else
            bc=[bc,diff(t_segment)];
        end
    end
end
%----- END OF CODE -----

function stageCost=L(x,xr,u,ur,p,t,vdat)
% L - Returns the stage cost. Warp function
%----- BEGIN CODE -----
if isfield(vdat,'Xscale')
    x=scale_variables_back( x, vdat.Xscale, vdat.Xshift );
    if ~isempty(xr)
        xr=scale_variables_back( xr, vdat.Xscale, vdat.Xshift );
    end
    u=scale_variables_back( u, vdat.Uscale, vdat.Ushift );
    if isfield(vdat,'Pscale')
        p=scale_variables_back( p, vdat.Pscale, vdat.Pshift );
    end
    if ~isempty(ur)
        ur=scale_variables_back( ur, vdat.Uscale, vdat.Ushift );
    end
    if strcmp(vdat.mode.currentMode,'Feasibility')
        stageCost=0*t;
    else
        stageCost=L_unscaled(x,xr,u,ur,p,t,vdat);
    end
else
    if strcmp(vdat.mode.currentMode,'Feasibility')
        stageCost=0*t;
    else
        stageCost=L_unscaled(x,xr,u,ur,p,t,vdat);
    end
end
%----- END OF CODE -----

```

```

function boundaryCost=E(x0,xf,u0,uf,p,t0,tf,vdat)
% E - Returns the boundary value cost. Warp function
%----- BEGIN CODE -----
if isfield(vdat,'Xscale')
    x0=scale_variables_back( x0', vdat.Xscale, vdat.Xshift );
    xf=scale_variables_back( xf', vdat.Xscale, vdat.Xshift );
    u0=scale_variables_back( u0', vdat.Uscale, vdat.Ushift );
    uf=scale_variables_back( uf', vdat.Uscale, vdat.Ushift );
    if isfield(vdat,'Pscale')
        p=scale_variables_back( p', vdat.Pscale, vdat.Pshift );
    end
    if strcmp(vdat.mode.currentMode,'Feasibility')
        boundaryCost=sum(sum(p(:,end-vdat.mode.np*2+1:end)));
    else
        boundaryCost=E_unscaled(x0,xf,u0,uf,p,t0,tf,vdat);
    end
else
    if strcmp(vdat.mode.currentMode,'Feasibility')
        boundaryCost=sum(sum(p(:,end-vdat.mode.np*2+1:end)));
    else
        boundaryCost=E_unscaled(x0,xf,u0,uf,p,t0,tf,vdat);
    end
end
%----- END OF CODE -----

function dx = f(x,u,p,t,vdat)
% f - Returns the ODE right hand side where x'= f(x,u,p,t). Warp
function
%----- BEGIN CODE -----
if isfield(vdat,'Xscale')
    x=scale_variables_back( x, vdat.Xscale, vdat.Xshift );
    u=scale_variables_back( u, vdat.Uscale, vdat.Ushift );
    if isfield(vdat,'Pscale')
        p=scale_variables_back( p, vdat.Pscale, vdat.Pshift );
    end
    dx = f_unscaled(x,u,p,t,vdat);
    dx= scale_variables( dx, vdat.Xscale, 0 );
else
    dx = f_unscaled(x,u,p,t,vdat);
end
%----- END OF CODE -----

function c=g(x,u,p,t,vdat)
% g - Returns the path constraint function where gl <= g(x,u,p,t)
<= gu. Warp function
%----- BEGIN CODE -----
if isfield(vdat,'Xscale')
    x=scale_variables_back( x, vdat.Xscale, vdat.Xshift );
    u=scale_variables_back( u, vdat.Uscale, vdat.Ushift );
    if isfield(vdat,'Pscale')

```



```

        p=scale_variables_back( p, vdat.Pscale, vdat.Pshift );
    end
    c = g_unscaled(x,u,p,t,vdat);
else
    c = g_unscaled(x,u,p,t,vdat);
end

if isfield(vdat,'gFilter')
    c(:,vdat.gFilter)=[];
end

if strcmp(vdat.mode.currentMode,'Feasibility')
    c=[c-p(:,end-vdat.mode.np*2+1:end-vdat.mode.np) c+p(:,end-
vdat.mode.np+1:end)];
end
%----- END OF CODE -----

function cr=avrc(x,u,p,t,data)
% avrc - Returns the rate constraint algebraic function where
% [xrl url] =< avrc(x,u,p,t) =< [xru uru]
% The function must be vectorized and xi, ui, pi are column
% vectors taken as x(:,i), u(:,i) and p(:,i).
% Each constraint corresponds to one column of c
%
% Syntax:  cr=avrc(x,u,p,t,data)
% Output:  cr - constraint function
%----- BEGIN CODE -----
[ cr ] = addRateConstraint( x,u,p,t,data );
%----- END OF CODE -----

function bc=b(x0,xf,u0,uf,p,t0,tf,vdat,varargin)
% b - Returns a column vector containing the evaluation of the
% boundary constraints: bl =< bf(x0,xf,u0,uf,p,t0,tf) =< bu. Warp
% function
%----- BEGIN CODE -----
bc=b_unscaled(x0,xf,u0,uf,p,t0,tf,vdat,varargin);
if isfield(vdat,'Xscale') % isfield - Determine whether input
is structure array field
    if ~isempty(bc)
        x0=scale_variables_back( x0', vdat.Xscale, vdat.Xshift );
        xf=scale_variables_back( xf', vdat.Xscale, vdat.Xshift );
        u0=scale_variables_back( u0', vdat.Uscale, vdat.Ushift );
        uf=scale_variables_back( uf', vdat.Uscale, vdat.Ushift );
        if isfield(vdat,'Pscale')
            p=scale_variables_back( p', vdat.Pscale, vdat.Pshift
);
        end
        bc=b_unscaled(x0,xf,u0,uf,p,t0,tf,vdat,varargin);
    end
end
%----- END OF CODE -----

```

B.3 Код общей настройки

```
function options = settings(N)
% General settings (0 => no and 1 => yes)

%%%----- BEGIN CODE -----%%%
%% Discreti Method:
%-----
options.transcription='euler';
%% Result Representation:
%-----
options.resultRep='default';
%% Derivative generation :
%-----
options.derivatives='numeric';
% Numeric generation of the Hessian:
%-----
-
options.hessianFD='central';
options.perturbation.H=[]; % Perturbation size for the second
derivatives
options.perturbation.J=[]; % Perturbation size for the first
derivatives

%% NLP solver
%-----
% Cold/Warm/Hot Start
options.start='Cold';
% Automatic scaling
options.scaling=1;
% Output settings
% Display computation time
options.print.time=1;
% Display relative local discretization error
options.print.relative_local_error=1;
% Display cost
options.print.cost=1;
% Plot states
options.plot.states=1;
% Plot inputs
options.plot.inputs=1;
% Plot Lagrange multipliers
options.plot.multipliers=1;
options.NLPsolver='ipopt';
% Desired convergence tolerance (relative). The default value is
1e-8.
options.ipopt.tol=1e-4;
% Print level. The valid range for this integer option is [0,12]
and its default value is 5.
options.ipopt.print_level=5;
```

```

% Maximum number of iterations. The default value is 3000.
options.ipopt.max_iter=3000;
options.ipopt.print_timing_statistics='yes';
% Determines which barrier parameter update strategy is to be
used.
options.ipopt.mu_strategy='adaptive';
% Indicates what information for the Hessian of the Lagrangian
function is
options.ipopt.hessian_approximation='limited-memory';
% Maximum size of the history for the limited quasi-Newton
Hessian approximation. The valid range for this integer option
is [0, +inf) and its default value is 6.
options.ipopt.limited_memory_max_history=6;
% Threshold for successive iterations where update is skipped
for the quasi-Newton approximation. The valid range for this
integer option is [1,+inf) and its default value is 2.
options.ipopt.limited_memory_max_skipping=1;

%% Direct transcription settings
%-----
% Number of integration nodes in the interval t=[0,tf];
nodes=steps+1.
options.nodes=N;

% Adaptively spaced segments
options.adaptseg=0;

% Minimum time interval
options.mintimeinterval=0.001;

% Distribution of integration steps. Set tau=0 for equispaced
steps.
options.tau=0;

%%----- END OF CODE -----%%

```

V.3 Код формирующий траекторий на модели земли

Этот код требуется пользоваться версией МАТЛАБ - 2018а и выше, чтобы получить ожидаемые результаты.

```

% World Map
%%----- BEGIN CODE -----%%
% load coastlines
% [latcells, loncells] = polysplit(coastlat, coastlon);
% plotm(coastlat, coastlon)

%% 2D Part of the Earth
% Make a map of the geoid over South America and the central
Pacific.

```

```

figure
%%% Profile A
% worldmap([-80 80],[-170 170])
%%% Profile B
worldmap([-80 80],[-135 225])
% load geoid
geoshow('landareas.shp','FaceColor',[0.4 1 0.4])
% geoshow(geoid, geoidrefvec, 'DisplayType', 'texturemap');
load coastlines
% geoshow(coastlat,coastlon)
lat =
speval(solution.Xp,2,[solution.T(:,1);solution.tf])*180/pi;
lon =
speval(solution.Xp,3,[solution.T(:,1);solution.tf])*180/pi;
plotm(lat,lon,'rp-'); grid on

%%% plot points of latitude and longitude
hold on
plotm(lat(1), lon(1), '.k', 'MarkerSize',30)
plotm(lat(end), lon(end), '.k', 'MarkerSize',30)
% text(lat(1)+5, lon(1)+5,'A')
% text(lat(end)+5, lon(end)+5,'B')
hold off
%%%----- END OF CODE -----%%%

```

V.4 Дополнительные строки при наличии угла скольжения

Эти строки необходимо добывать в коде (MinFuelFlight) при использовании угла скольжения в качестве параметра управления.

```

%--Initial Conditions--          --Terminal Conditions--
beta_0      = 0.0*pi/180;

% variable bounds
%-----MIN-----              -----MAX-----
beta_min = -10*pi/180;          beta_max=10*pi/180;

% Guess the input sequences with [u0 uf]
guess.inputs(:,1)=[alpha0 alpha0 alpha0];
guess.inputs(:,2)=[phi_0 phi_0 phi_0];
guess.inputs(:,3)=[throttle_0 throttle_0 throttle_0];
guess.inputs(:,4)=[beta_0 beta_0 beta_0];

function dx = f_unscaled(x,u,p,t,data)
alpha    = u(:,1);
phi      = u(:,2);
throttle = u(:,3);
beta     = u(:,4);

```

```

Y = 0.1994.*V.^2.*(+0.0000 - 0.1038.*beta + 0.0299.*beta.^2); %
Side-Force
%% equations of motions
gamma_dot = (1./V).*((L.*cos(phi) - cy*Y.*sin(phi) -
m.*auxdata.g.*cos(gamma))./(m)
+(V.^2.*cos(gamma))./(H+auxdata.rearth)+2.*auxdata.omegae.*V.*cos
(PPH).*cos(chi)+auxdata.omegae.^2.*(H+auxdata.rearth).*cos(PPH).^
2.*(cos(gamma)+sin(gamma).*tan(PPH).*sin(chi)));
chi_dot = (1./V).*((L.*sin(phi) +
cy*Y.*cos(phi))./(cos(gamma).*m) -
(V.^2./ (H+auxdata.rearth)).*cos(gamma).*cos(chi).*tan(PPH)+2.*aux
data.omegae.*V.*(tan(gamma).*cos(PPH).*sin(chi)-sin(PPH))-
(auxdata.omegae.^2.*(H+auxdata.rearth)./cos(gamma)).*sin(PPH).*co
s(PPH).*cos(chi));

```

В.5 Дополнительные коды

Эти коды являются более 20 разных кодов, они нужны для расчёта функции и матрикса Лагранжа и Гессияна, и других функций в этой работе. Эти коды загружали в Интернете, вы сможете скачать через эту ссылку (вниз), или собираете сам из сайта МАТЛАБ и/или сайта GitHub.

<https://bit.ly/2Et4ps9>

Приложение Г: уравнения задачи Понтрягина

В качестве функции параметров управления, рассмотрим программы угла атаки $\alpha(t)$ и угла скольжения $\beta(t)$.

Напишем параметры движения (параметры состояния) от системы уравнений динамики 1.1-1.5:

$$x = \{V, \gamma, \chi, p^t, q^t, r^t, \phi', \theta', \psi', h, \theta, \phi, v\} = \{V, \Omega, \Lambda, \Phi, v\} \quad (\text{Г.1})$$

Где:

$$V = \{V, \gamma, \chi\} \quad (\text{Г.3})$$

$$\Omega = \{p^t, q^t, r^t\} \quad (\text{Г.4})$$

$$\Lambda = \{\phi', \theta', \psi'\} \quad (\text{Г.5})$$

$$\Phi = \{h, \theta, \phi\} \quad (\text{Г.6})$$

V – Линейные Скорости, Ω – угловые Скорости, Λ – Углы Эйлера, Φ – Расположение в координатах навигации и v – Расход топлива.

Система уравнений Г.1 является 13 параметров входят в 13 уравнений динамики; Для определения оптимального закона для угла атаки можно пользоваться принцип максимума аэродинамического качества K [1], но этот принцип не помогает в ситуациях полёта с углом скольжения и минимизации расхода топлива. Будем решать нашу задачу численно, где после формулировки необходимых уравнений и ограничений (Гамильтониана, уравнений динамики (уже оформлены), уравнений совместного состояния, условий оптимальности, граничных условий, ограничения пути) пользуемся методом конечных разностей для решения систем новых уравнений [11].

Командное управление – найти ориентации аппарата в любой момент времени, это определяет направление тяги и аэродинамические силы в процессе управления аппаратом. Для того требуется начальные условия $\{V_0, \Omega_0, \Lambda_0, \Phi_0, m_0\}$ и конечные условия $\{V_f, \Omega_f, \Lambda_f, \Phi_f, m_f\}$.

Целью оптимального управления является создание воздействия на систему u , для движения системы по траектории x , так что индекс производительности J сводится к минимуму.

Для удобства работы перепишем систему уравнения движения в форме обычных дифференциальных уравнений $\dot{x}(t) = f(x, u, t)$, где x называется параметром состояния, u - функцией управления, выбранной из параметров динамики; берём $u = \alpha$ в продольном управлении, $u = \beta$ в боковом управлении и $u = \Gamma$ в поперечном управлении, где Γ - угол крена.

Новая система уравнения 1.19, решается с помощью дополнительных уравнений 1.6-1.11.

В процессе оптимизации будем искать функции управлений (программу управления) продольного движения и бокового движения поэтому чтобы облегчить задачу оптимизации разделяем полную систему движения самолёты (Г.7) на двух системы; продольное движение (Г.8) и боковое движение (Г.9).

$$\left. \begin{array}{l} \text{Уравнения (1.1)} \\ \dot{q}^t = \frac{M}{I_y} \\ \dot{\theta}' = q^t + \frac{V \cos(\gamma) \cos(\chi)}{R} \\ \dot{h} = V \sin(\gamma) \\ \dot{\theta} = V \cos(\gamma) \sin(\chi) / (R \cos(\phi)) \\ \dot{\phi} = V \cos(\gamma) \cos(\chi) / R \\ \dot{m} = -v \end{array} \right\} \quad (\text{Г.8})$$

Уравнения бокового движения:

$$\begin{aligned}
\frac{dV}{dt} &= \frac{T_x - D}{m} - g \sin(\gamma) + \omega^2 R \cos^2(\phi) (\sin(\gamma) - \cos(\gamma) \tan(\phi) \sin(\chi)) \\
\frac{d\gamma}{dt} &= \frac{1}{V} \left(\frac{T_z + L \cos(\mu_a) - Y_a \sin(\mu_a)}{m} - g \cos(\gamma) + \frac{V^2}{r} \cos(\gamma) + 2\omega V \cos(\phi) \cos(\chi) + \omega^2 R \cos^2(\phi) (\cos(\gamma) + \sin(\gamma) \tan(\phi) \sin(\chi)) \right) \\
\frac{d\chi}{dt} &= \frac{1}{V} \left(\frac{T_y + L \sin(\mu_a) + Y_a \cos(\mu_a)}{m \cos(\gamma)} - \frac{V^2}{r} \cos(\gamma) \cos(\chi) \tan(\phi) + 2\omega V (\tan(\gamma) \cos(\phi) \sin(\chi) - \sin(\phi)) - \frac{\omega^2 R}{\cos(\gamma)} \sin(\phi) \cos(\phi) \cos(\chi) \right) \\
\dot{p}^t &= \frac{p^t q^t (I_x I_{xz} - I_y I_{xz}) + q^t r^t (I_y I_z - I_{xz}^2 - I_z^2) + L I_z + N I_{xz}}{I_x I_z - I_{xz}^2} \\
\dot{q}^t &= \frac{-r^t p^t \cdot (I_x - I_z) - (p^{t^2} - r^{t^2}) \cdot I_{zx} + M}{I_y} \\
\dot{r}^t &= \frac{-p^t q^t \cdot (I_y - I_x) + I_{xz} \cdot (\dot{p}^t - q^t r^t) + N}{I_z} \\
\dot{\phi}' &= p^l + q^l (\sin(\phi') \tan(\theta')) + r^l (\cos(\phi') \tan(\theta')) \\
\dot{\theta}' &= q^l \cos(\phi') - r^l \sin(\phi') \\
\dot{\psi}' &= q^l \left(\frac{\sin(\phi')}{\cos(\theta')} \right) + r^l \left(\frac{\cos(\phi')}{\cos(\theta')} \right) \\
\dot{h} &= V \sin(\gamma) \\
\dot{\theta} &= V \cos(\gamma) \sin(\chi) / (R \cos(\phi)) \\
\dot{\phi} &= V \cos(\gamma) \cos(\chi) / R \\
\dot{m} &= -v
\end{aligned}
\tag{F.7}$$

$$\left. \begin{aligned}
 &\text{Уравнения (1.1)} \\
 &\dot{p}^t = \frac{LI_z + NI_{xz}}{I_x I_z - I_{xz}^2} \\
 &\dot{r}^t = \frac{N(I_x I_z - I_{xz}^2) + I_{xz}(LI_z + NI_{xz})}{I_z(I_x I_z - I_{xz}^2)} \\
 &\dot{\phi}' = p^l(p^t, \psi', \phi, V, \chi, R) \\
 &\dot{\psi}' = q^l(q^t, \psi', \phi', \phi, V, \chi, R) \cdot \sin(\phi') + r^l(q^t, \psi', \phi', \phi, V, \chi, R) \cdot \cos(\phi') \\
 &\dot{\theta} = V \cos(\gamma) \sin(\chi) / (R \cos(\phi)) \\
 &\dot{\phi} = V \cos(\gamma) \cos(\chi) / R \\
 &\dot{m} = -v
 \end{aligned} \right\} \quad (\Gamma.9)$$

Система уравнения (Г.9) решается с помощью уравнений 1.6-1.10 при замене $\theta' = 0$.

Функция обычных дифференциальных уравнений, определяющая состояние самолёта (Г.7), можно переписаться в векторной форме:

$$\dot{\mathbf{x}} = f(\mathbf{x}, u, t) \quad (\Gamma.10)$$

Где $\mathbf{x} = \{V, \gamma, \chi, q^t, \theta', h, \phi, \theta, m\}$ и $u = \alpha$ в продольном движении, $\mathbf{x} = \{V, \gamma, \chi, r^t, q^t, \phi', \psi', \theta, \phi, m\}$ и $u = \beta$ в боковом движении.

Напомним задачу оптимизации – найти программу управления u , который минимизирует расход топлива v , где минимизация \dot{v} достигается при максимизации функцию J , т.е. при: $J = -v_f$ т.е. $Max(J) = -Min(v_f)$. Берём угол атаки $u = \alpha$ для программы управления продольного движения, и угол скольжения $u = \beta$ для программы управления бокового движения.

Г.1 Программа управления угла атаки

Г.1.1 Оформление функцию Гамильтониана

На базе уравнений движения (Г.8), функцию Гамильтона можно обозначить как:

$$\begin{aligned}
H = & \psi_V \left(\frac{T_y - D}{m} - g \sin(\gamma) + \omega^2 R \cos^2(\phi) (\sin(\gamma) - \cos(\gamma) \tan(\phi) \sin(\chi)) \right) \\
& + \psi_\gamma \left(\frac{1}{V} \left(\frac{L \cos(\mu_a) - Y_a \sin(\mu_a)}{m} - g \cos(\gamma) + \frac{V^2}{R} \cos(\gamma) + 2\omega V \cos(\phi) \cos(\chi) \right. \right. \\
& \left. \left. + \omega^2 R \cos^2(\phi) (\cos(\gamma) + \sin(\gamma) \tan(\phi) \sin(\chi)) \right) \right) \\
& + \psi_\chi \left(\frac{1}{V} \left(\frac{L \sin(\mu_a) + Y_a \cos(\mu_a)}{m \cos(\gamma)} - \frac{V^2}{R} \cos(\gamma) \cos(\chi) \tan(\phi) \right. \right. \\
& \left. \left. + 2\omega V (\tan(\gamma) \cos(\phi) \sin(\chi) - \sin(\phi)) - \frac{\omega^2 R}{\cos(\gamma)} \sin(\phi) \cos(\phi) \cos(\chi) \right) \right) \\
& + \psi_{q^t} \left(\frac{M}{I_y} \right) + \psi_{\theta'} \left(q^t + \frac{V \cos(\gamma) \cos(\chi)}{R} \right) + \psi_h (V \sin(\gamma)) \\
& + \psi_\phi \left(\frac{V \cos(\gamma) \cos(\chi)}{R} \right) + \psi_\theta \left(\frac{V \cos(\gamma) \sin(\chi)}{R \cos(\phi)} \right) - \psi_m v \quad (\Gamma. 11)
\end{aligned}$$

где $\psi_V, \psi_\gamma, \psi_\chi, \psi_{q^t}, \psi_{\theta'}, \psi_h, \psi_\phi, \psi_\theta$ векторы совместного состояния (сопряжённые переменные), ассоциированные с состояниями полной скорости V , углов наклона траектории γ и χ , вращательной скорости q^t , углов Эйлера θ' и положения самолёта h, θ, ϕ , соответственно. а ψ_m скаляр совместного состояния, ассоциированный с массой.

Система дифференциальных уравнений для $\psi_V, \psi_\gamma, \psi_\chi, \psi_{q^t}, \psi_{\theta'}, \psi_h, \psi_\phi, \psi_\theta, \psi_m$ имеет вид:

$$\dot{\psi}_x = -\frac{\partial H}{\partial x} \quad (\Gamma. 12)$$

Где \mathbf{x} – Параметры системы продольного движения.

Оптимальное решение должно удовлетворить конечное условие (Г.13), [14], и условие трансверсальности (Г.14):

$$\psi(V_f, \gamma_f, \chi_f, q^t_f, \theta'_f, h_f, \phi_f, \theta_f, m_f) = 0 \quad (\Gamma. 13)$$

где $V_f = V(t_f)$, $\gamma_f = \gamma(t_f)$, $\chi_f = \chi(t_f)$, $p_f^t = p^t(t_f)$, $\theta'_f = \theta'(t_f)$, $h_f = h(t_f)$, $\phi_f = \phi(t_f)$, $\theta_f = \theta(t_f)$ и $m_f = m(t_f)$.

$$\left([1 - H(t)]\delta t + \psi_V \delta V(t) + \psi_\gamma \delta \gamma(t) + \psi_\chi \delta \chi(t) + \psi_{q^t} \delta q^t(t) + \psi_{\theta'} \delta \theta'(t) + \psi_h \delta h(t) + \psi_\phi \delta \phi(t) + \psi_\theta \delta \theta(t) + \psi_m \delta m(t) \right)_i^f = 0 \quad (\Gamma.14)$$

где $\delta t, \delta V(t), \delta \gamma(t), \delta \chi(t), \delta q^t(t), \delta \theta'(t), \delta h(t), \delta \phi(t), \delta \theta(t), \delta m(t)$ — вариации соответствующих величин при времени $t \in [t_i, t_f]$.

Г.1.2 Уравнения совместного состояния

Уравнения совместного состояния могут быть получены от функции Гамильтона

Г.11:

$$\begin{aligned} \dot{\psi}_V = -\frac{\partial H}{\partial V} = & -\psi_V \left(\frac{1}{m} \frac{\partial}{\partial V} (T_y - D) \right) \\ & - \psi_\gamma \left(\frac{1}{mV} \frac{\partial}{\partial V} (LC\mu a) + \frac{C\gamma}{R} \right. \\ & \left. + \frac{1}{V^2} \left(gC\gamma - R\omega^2 C\phi^2 (S\gamma T\phi S\chi + C\gamma) + \frac{-LC\mu a + Y_a S\mu a}{m} \right) \right) \\ & - \psi_\chi \left(-\frac{C\gamma T\phi C\chi}{R} + \frac{1}{V^2} \left(-\frac{(LS\mu a)}{mC\gamma} + \frac{\omega^2 R S\phi C\phi C\chi}{C\gamma} \right) \right) - \psi_{q^t} \left(\frac{1}{I_y} \frac{\partial}{\partial V} M \right) \\ & - \psi_{\theta'} \left(\frac{C\gamma C\chi}{R} \right) - \psi_h (S\gamma) - \psi_\phi \left(\frac{C\gamma C\chi}{R} \right) - \psi_\theta \left(\frac{C\gamma S\chi}{RC\phi} \right) \quad (\Gamma.15) \end{aligned}$$

$$\dot{\psi}_{q^t} = -\frac{\partial H}{\partial q^t} = -\psi_{\theta'} \left(\frac{VC\gamma S\chi}{R} \right) \quad (\Gamma.15)$$

$$\dot{\psi}_{\theta'} = -\frac{\partial H}{\partial \theta'} = 0 \quad (\Gamma.16)$$

$$\dot{\psi}_\theta = -\frac{\partial H}{\partial \theta} = 0 \quad (\Gamma.17)$$

$$\begin{aligned}
\psi_\gamma = -\frac{\partial H}{\partial \gamma} = & -\psi_V(-gC\gamma + R\omega^2 C\phi^2(S\gamma T\phi S\chi + C\gamma)) \\
& -\psi_\gamma\left(\frac{R\omega^2 C\phi^2}{V}(-S\gamma + C\gamma T\phi S\chi) + S\gamma\left(\frac{g}{V} - \frac{V}{R}\right)\right) \\
& -\psi_\chi\left(\frac{-R\omega^2 S\phi C\phi C\chi S\gamma}{VC\gamma^2} + 2\omega C\phi S\chi(T\gamma^2 + 1) + \frac{VS\gamma C\chi T\phi}{R}\right) + \psi_{\theta'}\left(\frac{VS\gamma C\chi}{R}\right) \\
& -\psi_h(VC\gamma) + \psi_\phi\left(\frac{VS\gamma C\chi}{R}\right) + \psi_\theta\left(\frac{VS\gamma S\chi}{RC\phi}\right) \quad (\Gamma.18)
\end{aligned}$$

$$\begin{aligned}
\psi_\chi = -\frac{\partial H}{\partial \chi} = & \psi_V(R\omega^2 C\phi^2 C\gamma T\phi C\chi) - \psi_\gamma\left(-2\omega C\phi S\chi + \frac{R\omega^2}{V}C\phi^2 S\gamma T\phi C\chi\right) \\
& -\psi_\chi\left(\frac{V}{R}C\gamma S\chi T\phi + 2\omega T\gamma C\phi C\chi + \frac{\omega^2 R}{VC\gamma}S\phi C\phi S\chi\right) + \psi_{\theta'}\left(\frac{VC\gamma S\chi}{R}\right) + \psi_\phi\left(\frac{VC\gamma S\chi}{R}\right) \\
& -\psi_\theta\left(\frac{VC\gamma C\chi}{RC\phi}\right) \quad (\Gamma.19)
\end{aligned}$$

$$\begin{aligned}
\psi_\phi = -\frac{\partial H}{\partial \phi} = & -\psi_V(R\omega^2(-C\gamma S\phi^2 S\chi + 2C\gamma S\chi S\phi^2 - C\gamma C\phi^2 S\chi - S\gamma S2\phi)) \\
& -\psi_\gamma\left(-2\omega S\phi C\chi\right. \\
& \left. + \frac{R\omega^2}{V}(-2C\phi S\phi C\gamma - 2C\phi S\phi S\gamma T\phi S\chi + C\phi^2 S\gamma S\chi + S\phi^2 S\gamma S\chi)\right) \\
& -\psi_\chi\left(\frac{-VC\gamma C\chi(1 + T\phi^2)}{R} - 2\omega(T\gamma S\phi S\chi + C\phi) + \frac{R\omega^2 C\chi}{VC\gamma}(S\phi^2 - C\phi^2)\right) \\
& + \psi_\theta\left(\frac{VC\gamma S\chi S\phi}{RC\phi^2}\right) \quad (\Gamma.21)
\end{aligned}$$

$$\psi_m = -\frac{\partial H}{\partial m} = -\psi_V\left(\frac{-T_y + D}{m^2}\right) - \psi_\gamma\left(\frac{-LC\mu a}{Vm^2}\right) \quad (\Gamma.22)$$

$$\begin{aligned}
\psi_h = -\frac{\partial H}{\partial h} = & -\psi_V(\omega^2 C\phi^2(-C\gamma T\phi S\chi + S\gamma)) - \psi_\gamma\left(-\frac{VC\gamma}{R^2} + \frac{\omega^2 C\phi^2}{V}(C\gamma + S\gamma T\phi S\chi)\right) \\
& -\psi_\chi\left(\frac{V}{R^2}C\gamma C\chi T\phi - \frac{\omega^2}{VC\gamma}S\phi C\phi C\chi\right) + \psi_{\theta'}\left(\frac{VC\gamma C\chi}{R^2}\right) + \psi_\phi\left(\frac{VC\gamma C\chi}{R^2}\right) \\
& + \psi_\theta\left(\frac{VC\gamma C\chi}{R^2 C\phi}\right) \quad (\Gamma.20)
\end{aligned}$$

Здесь произведения считали с помощью программного обеспечения МАБЛЬ [15] для символьных расчёт. Производные $\frac{\partial}{\partial V}(T_y - D)$, $\frac{\partial}{\partial V}(LC\mu a - Y_a S\mu a)$, $\frac{\partial}{\partial V}(LS\mu a + Y_a C\mu a)$ и $\frac{\partial}{\partial V}M$ получаются от аэродинамических полиномов, формирующийся на базе аэродинамических таблиц.

Г.1.3 Условие оптимальности

Условие максимумы функции H по управлению угла атаки α имеет вид:

$$\frac{\partial H}{\partial \alpha} = 0 \quad (\text{а}), \quad \frac{\partial^2 H}{\partial \alpha^2} \leq 0 \quad (\text{б}) \quad (\text{Г.23})$$

Вычисление дифференциалы и аэродинамической силы дает следующие результаты:

$$\begin{aligned} \frac{\partial H}{\partial \alpha} &= \psi_V \left(\frac{1}{m} \left(-\frac{\partial D}{\partial \alpha} \right) \right) + \psi_Y \left(\frac{1}{mV} \frac{\partial (LC\mu a)}{\partial \alpha} \right) + \psi_X \left(\frac{1}{mC_Y V} \frac{\partial (LS\mu a)}{\partial \alpha} \right) + \psi_{q^t} \left(\frac{1}{I_y} \frac{\partial M}{\partial \alpha} \right) = 0 \\ f_1 = \frac{\partial H}{\partial \alpha} &= -\psi_V \left(\frac{\partial C_D}{\partial \alpha} \right) + \psi_Y \left(\frac{C\mu a}{V} \frac{\partial C_L}{\partial \alpha} \right) + \psi_X \left(\frac{S\mu a}{C_Y V} \frac{\partial C_L}{\partial \alpha} \right) + \psi_{q^t} \left(\frac{mL_{\text{этал}}}{I_y} \frac{\partial C_m}{\partial \alpha} \right) = 0 \quad (\text{Г.24}) \end{aligned}$$

От условия Г.23.б напомним:

$$\frac{\partial^2 H}{\partial \alpha^2} = -\psi_V \left(\frac{\partial^2 C_D}{\partial \alpha^2} \right) + \psi_Y \left(\frac{C\mu a}{V} \frac{\partial^2 C_L}{\partial \alpha^2} \right) + \psi_X \left(\frac{S\mu a}{C_Y V} \frac{\partial^2 C_L}{\partial \alpha^2} \right) + \psi_{q^t} \left(\frac{mL_{\text{этал}}}{I_y} \frac{\partial^2 C_m}{\partial \alpha^2} \right) < 0 \quad (\text{Г.25})$$

От Г.25 и при $\mu a = 0$ имеем $\frac{\partial^2 C_D}{\partial \alpha^2} > 0$, $\frac{\partial^2 C_L}{\partial \alpha^2} > 0$, $\frac{\partial^2 C_m}{\partial \alpha^2} < 0$, $V > 0$ и $\frac{mL_{\text{этал}}}{I_y} > 0$ это значит что $\psi_V, \psi_Y, \psi_{q^t}$ требуется иметь следующие значения $\psi_V > 0, \psi_Y < 0$ и $\psi_{q^t} > 0$ чтобы обеспечить максимуму функции H .

Условие Г.25 должно удовлетворяться в каждом шаге времени.

Г.2 Программа управления угла скольжения

Г.2.1 Оформление функцию Гамильтониана

На базе уравнений движения (Г.9), функцию Гамильтона можно обозначить в Г.26. Где $\psi_V, \psi_\gamma, \psi_\chi, \psi_{p^t}, \psi_{r^t}, \psi_{\phi'}, \psi_{\psi'}, \psi_\phi, \psi_\theta$ векторы совместного состояния (сопряжённые переменные), ассоциированные с состояниями поступательной скорости v , вращательной скорости p^t и r^t , углов Эйлера ϕ' и ψ' и положения самолёта ϕ и θ , соответственно. а ψ_m скаляр совместного состояния, ассоциированный с массой.

Система дифференциальных уравнений для $\psi_V, \psi_\gamma, \psi_\chi, \psi_{p^t}, \psi_{r^t}, \psi_{\phi'}, \psi_{\psi'}, \psi_\phi, \psi_\theta, \psi_m$ имеет вид:

$$\dot{\psi}_x = -\frac{\partial H}{\partial x} \quad (\text{Г.27})$$

Где x – Параметры системы бокового движения.

Оптимальное решение должно удовлетворить конечное условие (Г.28), [14], и условие трансверсальности (Г.38):

$$\psi(V_f, \gamma_f, \chi_f, p_f^t, r_f^t, \phi_f', \psi_f', \theta_f, \phi_f, m_f) = 0 \quad (\text{Г.28})$$

где $V_f = V(t_f), \gamma_f = \gamma(t_f), \chi_f = \chi(t_f), p_f^t = p^t(t_f), r_f^t = r^t(t_f), \phi_f' = \phi'(t_f), \psi_f' = \psi'(t_f), \phi_f = \phi(t_f), \theta_f = \theta(t_f)$ и $m_f = m(t_f)$.

$$\begin{aligned}
H = & \psi_V \left(\frac{T_y - D}{m} - g \sin(\gamma) + \omega^2 R_E \cos^2(\phi) (\sin(\gamma) - \cos(\gamma) \tan(\phi) \sin(\chi)) \right) \\
& + \psi_\gamma \left(\frac{1}{V} \left(\frac{L \cos(\mu_a) - Y_a \sin(\mu_a)}{m} - g \cos(\gamma) + \frac{V^2}{R_E} \cos(\gamma) + 2\omega V \cos(\phi) \cos(\chi) \right. \right. \\
& \left. \left. + \omega^2 R \cos^2(\phi) (\cos(\gamma) + \sin(\gamma) \tan(\phi) \sin(\chi)) \right) \right) \\
& + \psi_\chi \left(\frac{1}{V} \left(\frac{L \sin(\mu_a) + Y_a \cos(\mu_a)}{m \cos(\gamma)} - \frac{V^2}{R_E} \cos(\gamma) \cos(\chi) \tan(\phi) \right. \right. \\
& \left. \left. + 2\omega V (\tan(\gamma) \cos(\phi) \sin(\chi) - \sin(\phi)) - \frac{\omega^2 R_E}{\cos(\gamma)} \sin(\phi) \cos(\phi) \cos(\chi) \right) \right) \\
& + \psi_{p^t} \left(\frac{L I_z + N I_{xz}}{I_x I_z - I_{xz}^2} \right) + \psi_{r^t} \left(\frac{N(I_x I_z - I_{xz}^2) + I_{xz}(L I_z + N I_{xz})}{I_z(I_x I_z - I_{xz}^2)} \right) \\
& + \psi_{\phi'} p^l(p^t, \psi', \phi, V, \chi) \\
& + \psi_{\psi'} (q^l(q^t, \psi', \phi', \phi, V, \chi) \cdot \sin(\phi') + r^l(r^t, \psi', \phi', \phi, V, \chi) \cdot \cos(\phi')) \\
& + \psi_\theta \left(\frac{V \cos(\gamma) \sin(\chi)}{R_E \cos(\phi)} \right) + \psi_\phi \left(\frac{V \cos(\gamma) \cos(\chi)}{R_E} \right) - \psi_m v \quad (\Gamma.26)
\end{aligned}$$

$$\begin{aligned}
& \left([1 - H(t)] \delta t + \psi_V \delta V(t) + \psi_\gamma \delta \gamma(t) + \psi_\chi \delta \chi(t) + \psi_{p^t} \delta p^t(t) + \psi_{r^t} \delta r^t(t) + \psi_{\phi'} \delta \phi'(t) \right. \\
& \left. + \psi_{\psi'} \delta \psi'(t) + \psi_\phi \delta \phi(t) + \psi_\theta \delta \theta(t) + \psi_m \delta m(t) \right)_i^f = 0 \quad (\Gamma.29)
\end{aligned}$$

где $\delta t, \delta V(t), \delta \gamma(t), \delta \chi(t), \delta p^t(t), \delta r^t(t), \delta \phi'(t), \delta \psi'(t), \delta \theta(t), \delta \phi(t), \delta m(t)$ — вариации соответствующих величин при времени $t \in [t_i, t_f]$.

Г.2.2 Уравнения совместного состояния

Уравнения совместного состояния могут быть получены от функции Гамильтона Г.26:

$$\begin{aligned}
\dot{\psi}_V = -\frac{\partial H}{\partial V} = & -\psi_V \left(\frac{1}{m} \frac{\partial}{\partial V} (T_y - D) \right) - \psi_\gamma \left(+\frac{C\gamma}{R} + \frac{1}{V^2} (+gC\gamma - R\omega^2 C\phi^2 (S\gamma T\phi S\chi + C\gamma)) \right) \\
& - \psi_\chi \left(\frac{1}{mC\gamma V} \frac{\partial}{\partial V} (Y_a C\mu a) - \frac{C\gamma T\phi C\chi}{R} + \frac{1}{V^2} \left(-\frac{(Y_a C\mu a)}{mC\gamma} + \frac{\omega^2 R S\phi C\phi C\chi}{C\gamma} \right) \right) \\
& - \psi_{p^t} \left(\frac{\frac{\partial L}{\partial V} I_z + \frac{\partial N}{\partial V} I_{xz}}{I_x I_z - I_{xz}^2} \right) - \psi_{r^t} \left(\frac{\frac{\partial N}{\partial V} (I_x I_z - I_{xz}^2) + I_{xz} \left(\frac{\partial L}{\partial V} I_z + \frac{\partial N}{\partial V} I_{xz} \right)}{I_z (I_x I_z - I_{xz}^2)} \right) \\
& - \psi_{\psi'} \left(\frac{C\gamma S\chi S\phi}{RC\phi} \right) + \psi_{\phi'} \left(\frac{C\psi' C\gamma S\chi}{R} \right) - \psi_\phi \left(\frac{C\gamma C\chi}{R} \right) - \psi_\theta \left(\frac{C\gamma S\chi}{RC\phi} \right) \quad (\Gamma.30)
\end{aligned}$$

$$\dot{\psi}_{p^t} = -\frac{\partial H}{\partial p^t} = -\psi_{\phi'} \quad (\Gamma.31)$$

$$\dot{\psi}_{r^t} = -\frac{\partial H}{\partial r^t} = -\psi_{\psi'} \cos(\phi') \quad (\Gamma.32)$$

$$\dot{\psi}_{\phi'} = -\frac{\partial H}{\partial \phi'} = -\psi_{\psi'} (q^t \cos(\phi') - r^t \sin(\phi')) \quad (\Gamma.33)$$

$$\dot{\psi}_{\psi'} = -\frac{\partial H}{\partial \psi'} = -\psi_{\phi'} \left(\omega S\psi' C\phi + \frac{VS\psi' C\gamma S\chi}{R} \right) \quad (\Gamma.34)$$

$$\dot{\psi}_\theta = -\frac{\partial H}{\partial \theta} = 0 \quad (\Gamma.35)$$

$$\dot{\psi}_m = -\frac{\partial H}{\partial m} = -\psi_V \left(\frac{-T_y + D}{m^2} \right) + \psi_\chi \left(\frac{Y_a C\mu a}{Vm^2 C\gamma} \right) \quad (\Gamma.36)$$

$$\begin{aligned}
\dot{\psi}_\gamma = -\frac{\partial H}{\partial \gamma} = & -\psi_V (-gC\gamma + R\omega^2 C\phi^2 (S\gamma T\phi S\chi + C\gamma)) \\
& - \psi_\gamma \left(\frac{R\omega^2 C\phi^2}{V} (-S\gamma + C\gamma T\phi S\chi) + S\gamma \left(\frac{g}{V} - \frac{V}{R} \right) \right) \\
& - \psi_\chi \left(\frac{-R\omega^2 S\phi C\phi C\chi S\gamma}{VC\gamma^2} + 2\omega C\phi S\chi (T\gamma^2 + 1) + \frac{S\gamma (Y_a C\mu a)}{VmC\gamma^2} + \frac{VS\gamma C\chi T\phi}{R} \right) \\
& + \psi_{\psi'} \left(\frac{VS\gamma C\chi S\phi}{RC\phi} \right) - \psi_{\phi'} \left(\frac{VC\psi' S\gamma S\chi}{R} \right) + \psi_\phi \left(\frac{VS\gamma C\chi}{R} \right) \\
& + \psi_\theta \left(\frac{VS\gamma S\chi}{RC\phi} \right) \quad (\Gamma.37)
\end{aligned}$$

$$\begin{aligned}
\dot{\psi}_\chi = -\frac{\partial H}{\partial \chi} = & \psi_V(R\omega^2 C\phi^2 C\gamma T\phi C\chi) - \psi_\gamma \left(-2\omega C\phi S\chi + \frac{R\omega^2}{V} C\phi^2 S\gamma T\phi C\chi \right) \\
& - \psi_\chi \left(\frac{V}{R} C\gamma S\chi T\phi + 2\omega T\gamma C\phi C\chi + \frac{\omega^2 R}{VC\gamma} S\phi C\phi S\chi \right) - \psi_{\psi'} \left(\frac{VC\gamma C\chi S\phi}{RC\phi} \right) \\
& + \psi_{\phi'} \left(\frac{VC\psi' C\gamma C\chi}{R} \right) + \psi_\phi \left(\frac{VC\gamma S\chi}{R} \right) - \psi_\theta \left(\frac{VC\gamma C\chi}{RC\phi} \right) \quad (\Gamma.38)
\end{aligned}$$

$$\begin{aligned}
\dot{\psi}_\phi = -\frac{\partial H}{\partial \phi} = & -\psi_V(R\omega^2(-C\gamma S\phi^2 S\chi + 2C\gamma C\phi T\phi S\chi S\phi - C\gamma C\phi^2 S\chi - 2C\phi S\gamma S\phi)) \\
& - \psi_\gamma \left(-2\omega S\phi C\chi \right. \\
& \left. + \frac{R\omega^2}{V}(-2C\phi S\phi C\gamma - 2C\phi S\phi S\gamma T\phi S\chi + C\phi^2 S\gamma S\chi + S\phi^2 S\gamma S\chi) \right) \\
& - \psi_\chi \left(\frac{-VC\gamma C\chi(1+T\phi^2)}{R} - 2\omega(T\gamma S\phi S\chi + C\phi) + \frac{R\omega^2 C\chi}{VC\gamma}(S\phi^2 - C\phi^2) \right) \\
& - \psi_{\psi'} \left(\omega C\phi + \frac{VC\gamma S\chi S\phi^2}{RC\phi^2} + \frac{VC\gamma S\chi}{R} \right) - \psi_{\phi'}(\omega C\psi' S\phi) - \psi_\theta \left(\frac{VC\gamma S\chi S\phi}{RC\phi^2} \right) \quad (\Gamma.39)
\end{aligned}$$

Здесь все произведения считали с помощью программного обеспечения МАБЛЬ [15] для символьных расчёт.

Г.2.3 Условие оптимальности

Условие максимумы функции H по управлению угла скольжения β имеет вид:

$$\frac{\partial H}{\partial \beta} = 0 \text{ (а)}, \quad \frac{\partial^2 H}{\partial \beta^2} \leq 0 \text{ (б)} \quad (\Gamma.40)$$

Вычисление дифференциалы и аэродинамической силы дает следующие результаты:

$$\begin{aligned}
\frac{\partial H}{\partial \beta} = & -\psi_\gamma \left(\frac{S\mu a}{mV} \frac{\partial Y_a}{\partial \beta} \right) + \psi_\chi \left(\frac{C\mu a}{VmC\gamma} \frac{\partial Y_a}{\partial \beta} \right) + \psi_{p^t} \left(\frac{1}{I_x I_z - I_{xz}^2} \left(\frac{\partial L}{\partial \beta} I_z + \frac{\partial N}{\partial \beta} I_{xz} \right) \right) \\
& + \psi_{r^t} \left(\frac{1}{I_z(I_x I_z - I_{xz}^2)} \left(\frac{\partial N}{\partial \beta} (I_x I_z - I_{xz}^2) + I_{xz} \left(\frac{\partial L}{\partial \beta} I_z + \frac{\partial N}{\partial \beta} I_{xz} \right) \right) \right) = 0
\end{aligned}$$

$$\begin{aligned}
& -\psi_\gamma \left(\frac{0.5\rho V^2 S_{\text{этал}} S_{\mu\alpha}}{mV} \frac{\partial C_y}{\partial \beta} \right) + \psi_\chi \left(\frac{0.5\rho V^2 S_{\text{этал}} C_{\mu\alpha}}{VmC_\gamma} \frac{\partial C_y}{\partial \beta} \right) \\
& + \psi_{p^t} \left(\frac{0.5\rho V^2 S_{\text{этал}} b_{\text{этал}}}{I_x I_z - I_{xz}^2} \left(I_z \frac{\partial C_l}{\partial \beta} + I_{xz} \frac{\partial C_n}{\partial \beta} \right) \right) \\
& + \psi_{r^t} \left(\frac{0.5\rho V^2 S_{\text{этал}} b_{\text{этал}}}{I_z (I_x I_z - I_{xz}^2)} \left(\frac{\partial C_n}{\partial \beta} (I_x I_z - I_{xz}^2) + I_{xz} \left(\frac{\partial C_l}{\partial \beta} I_z + \frac{\partial C_n}{\partial \beta} I_{xz} \right) \right) \right) = 0
\end{aligned}$$

После разделения на выражении $\frac{0.5\rho V^2 S_{\text{этал}}}{m}$:

$$\begin{aligned}
f2 = & -\psi_\gamma \left(\frac{S_{\mu\alpha}}{V} \frac{\partial C_y}{\partial \beta} \right) + \psi_\chi \left(\frac{C_{\mu\alpha}}{VC_\gamma} \frac{\partial C_y}{\partial \beta} \right) + \psi_{p^t} \left(\frac{mb_{\text{этал}}}{I_x I_z - I_{xz}^2} \left(I_z \frac{\partial C_l}{\partial \beta} + I_{xz} \frac{\partial C_n}{\partial \beta} \right) \right) \\
& + \psi_{r^t} \left(\frac{mb_{\text{этал}}}{I_z (I_x I_z - I_{xz}^2)} \left(\frac{\partial C_n}{\partial \beta} (I_x I_z - I_{xz}^2) + I_{xz} \left(\frac{\partial C_l}{\partial \beta} I_z + \frac{\partial C_n}{\partial \beta} I_{xz} \right) \right) \right) = 0 \quad (\Gamma.41)
\end{aligned}$$

От условия Г.40.б напомним:

$$\begin{aligned}
\frac{\partial^2 H}{\partial \beta^2} = & -\psi_\gamma \left(\frac{S_{\mu\alpha}}{V} \frac{\partial^2 C_y}{\partial \beta^2} \right) + \psi_\chi \left(\frac{C_{\mu\alpha}}{VC_\gamma} \frac{\partial^2 C_y}{\partial \beta^2} \right) + \psi_{p^t} \left(\frac{mb_{\text{этал}}}{I_x I_z - I_{xz}^2} \left(I_z \frac{\partial^2 C_l}{\partial \beta^2} + I_{xz} \frac{\partial^2 C_n}{\partial \beta^2} \right) \right) \\
& + \psi_{r^t} \left(\frac{mb_{\text{этал}}}{I_z (I_x I_z - I_{xz}^2)} \left(\frac{\partial^2 C_n}{\partial \beta^2} (I_x I_z - I_{xz}^2) + I_{xz} \left(\frac{\partial^2 C_l}{\partial \beta^2} I_z + \frac{\partial^2 C_n}{\partial \beta^2} I_{xz} \right) \right) \right) \\
< 0 \quad & (\Gamma.42)
\end{aligned}$$

Условие Г.42 должно удовлетворяться тоже в каждом шаге времени.

Уравнения Г.24 и Г.41 являются проблемой поиска корней (α и β), эти уравнения может быть решено численно с помощью модифицированного метода Ньютона:

$$\begin{pmatrix} \alpha_{\text{Опт}} \\ \beta_{\text{Опт}} \end{pmatrix}_{i+1} = \begin{pmatrix} \alpha_{\text{Опт}} \\ \beta_{\text{Опт}} \end{pmatrix}_i - \begin{bmatrix} \frac{\partial f1}{\partial \alpha} & \frac{\partial f1}{\partial \beta} \\ \frac{\partial f2}{\partial \alpha} & \frac{\partial f2}{\partial \beta} \end{bmatrix}^{-1} \begin{pmatrix} f1 \\ f2 \end{pmatrix} \quad (\Gamma.43)$$

При предположении что лонгитудинальные аэродинамические силы и моменты не зависят от угла скольжения β , и боковые аэродинамические силы и

моменты не зависят от угла атаки α . Это удовлетворительно при одном условии, угла крена равно нулю. Если угла крена равно нулю, тогда требуется дополнительные аэродинамические данные связывает продольные силы и моменты с β и боковые силы и моменты с α , на базе того добываем эти зависимости в уравнениях Г.25 и Г.42.

В уравнениях оптимизации α и β , f_1 зависит только от α , и f_2 зависит только от β т.е. $\frac{\partial f_1}{\partial \beta} = 0$ и $\frac{\partial f_2}{\partial \alpha} = 0$ и можно решить уравнения $\alpha_{\text{опт}}$ и $\beta_{\text{опт}}$ отдельно.