|  |  |
| --- | --- |
| Группа M3213 | К работе допущен |
| Студент \_Губанов Константин Романович | Работа выполнена |
| Преподаватель Хуснутдинова Наира Рустемовна | Отчет принят |

**Рабочий протокол и отчет по  
моделированию 1 “Лунолет”**

**Вариант 17**

1. Цель работы.

1.1 Построить графики зависимости вертикальной скорости Vy, ускорения ay, и высоты H от времени, чтобы визуализировать динамику движения аппарата.

1.2 Определить значение вертикальной скорости при достижении высоты 0 м (в момент посадки).

1. Задачи, решаемые при выполнении работы.

2.1 Рассчитать скорость, высоту и ускорение на каждом шаге времени, используя уравнение Мещерского, которое учитывает истечение продуктов сгорания и изменение массы аппарата.

2.2 Построить графики зависимости вертикальной скорости Vy, ускорения ay, и высоты H от времени для анализа параметров движения.

2.3 Определить вертикальную скорость аппарата в момент приземления на поверхность (когда высота становится равной нулю).

3. Объект исследования.

Движение лунного аппарата с учетом действия сил гравитации Луны и управляемого реактивного двигателя, работающего на керосине и жидком кислороде.

4. Метод экспериментального исследования.

Проведение численного моделирования движения аппарата по вертикальной оси с использованием уравнений движения, учитывающих изменение массы и сил, действующих на аппарат, с дальнейшей визуализацией результатов в виде графиков зависимости параметров движения от времени.

Моделирование производится при помощи кода, написанного на языке Python и использовании библиотек NumPy и Matplotlib.

5. Рабочие формулы и исходные данные.

**Исходные данные:**

gЛ =1.62 м/с2 - ускорение силы тяжести на Луне

M = 2150 кг - масса корабля 2000 кг, плюс пилот в скафандре 150 кг

Vp = 3660 м/с - скорость истечения продуктов сгорания

m = 150 кг - топливо и окислитель

= 15 кг/с - расход топлива двигателем

H0 = 2000м – начальная высота

V0y = 29 м/с – начальная вертикальная скорость

Vmax = 3 м/с – максимальная допустимая скорость при посадке

Уравнения кинематики (свободное падение):

Уравнение скорости:

Уравнение высоты:

Уравнения движения ракеты с переменной массой (уравнение Мещерского):

Уравнение тягового ускорения:

Уравнение полного ускорения:

Изменение параметров корабля при управляемой посадке:

Уравнение скорости:

Уравнение высоты:

Уравнение массы:

6. Код программы

Используемые библиотеки и исходные параметры

import numpy as np  
import matplotlib.pyplot as plt  
  
# Исходные параметры  
g\_L = 1.62 # ускорение на Луне, м/с^2  
M = 2150 # масса аппарата без топлива, кг  
m = 150 # начальная масса топлива, кг  
Vp = 3660 # скорость истечения продуктов сгорания, м/с  
m\_ = 15 # расход топлива, кг/с  
H0 = 2000 # начальная высота, м  
V0y = 29 # начальная вертикальная скорость, м/с (вниз положительная)  
Vmax = 3 # допустимая скорость при посадке, м/с  
dt = 0.1 # Шаг времени, с

Функция, моделирующая свободное падение аппарата с выключенным двигателем

def simulate\_free\_fall(H0, V0, gravity, dt):  
 time, height, velocity = [0], [H0], [V0]  
 while height[-1] > 0:  
 V = velocity[-1] + gravity \* dt  
 H = height[-1] - velocity[-1] \* dt - 0.5 \* gravity \* dt \*\* 2  
 time.append(time[-1] + dt)  
 velocity.append(V)  
 height.append(max(H, 0)) # Ограничиваем высоту до нуля  
 if H <= 0:  
 break  
 return np.array(time), np.array(height), np.array(velocity)

Функция, симулирующая движение аппарата при включенном двигателе (используется уравнение Мещерского)

def simulate\_powered\_descent(H0, V0, initial\_mass, gravity, exhaust\_velocity, fuel\_rate, max\_speed, dt):  
 time, height, velocity, mass, acceleration = [0], [H0], [V0], [initial\_mass], []  
 while height[-1] > 0 and mass[-1] > M:  
 m\_current = mass[-1]  
 thrust\_accel = -exhaust\_velocity \* fuel\_rate / m\_current  
 total\_accel = thrust\_accel - gravity  
 V = velocity[-1] + total\_accel \* dt  
 H = height[-1] - velocity[-1] \* dt - 0.5 \* total\_accel \* dt \*\* 2  
 m\_new = max(m\_current - fuel\_rate \* dt, M)  
 time.append(time[-1] + dt)  
 velocity.append(V)  
 height.append(max(H, 0))  
 mass.append(m\_new)  
 acceleration.append(total\_accel)  
 if H <= 0 or V <= max\_speed:  
 height[-1] = 0  
 break  
 return np.array(time), np.array(height), np.array(velocity), np.array(acceleration), np.array(mass)

Функция, которая вычисляет высоту, на которой нужно включить двигатель для безопасной посадки

def find\_engine\_start\_point(t\_free\_fall, H\_free\_fall, V\_free\_fall, base\_mass, fuel\_mass, dt):  
 for idx in reversed(range(len(t\_free\_fall))):  
 if H\_free\_fall[idx] <= 0:  
 continue  
 H\_engine\_on, V\_engine\_on = H\_free\_fall[idx], V\_free\_fall[idx]  
 total\_mass = base\_mass + fuel\_mass  
 t\_powered, H\_powered, V\_powered, a\_powered, m\_powered = simulate\_powered\_descent(  
 H\_engine\_on, V\_engine\_on, total\_mass, g\_L, Vp, m\_, Vmax, dt  
 )  
 if V\_powered[-1] <= Vmax:  
 return idx, t\_powered, H\_powered, V\_powered, a\_powered  
 return None, None, None, None, None

Функция, отвечающая за построение графиков

def plot\_results(time, height, velocity, acceleration):  
 plt.figure(figsize=(12, 8))  
  
 plt.subplot(3, 1, 1)  
 plt.plot(time, height, color='#f56642', label='Высота')  
 plt.title('Зависимость высоты от времени')  
 plt.xlabel('Время, с')  
 plt.ylabel('Высота, м')  
 plt.grid(True)  
 plt.legend()  
  
 plt.subplot(3, 1, 2)  
 plt.plot(time, velocity, color='#4299f5', label='Скорость')  
 plt.title('Зависимость вертикальной скорости от времени')  
 plt.xlabel('Время, с')  
 plt.ylabel('Скорость, м/с')  
 plt.grid(True)  
 plt.legend()  
  
 plt.subplot(3, 1, 3)  
 plt.plot(time[:-1], acceleration, color='#d442f5', label='Ускорение')  
 plt.title('Зависимость ускорения от времени')  
 plt.xlabel('Время, с')  
 plt.ylabel('Ускорение, м/с²')  
 plt.grid(True)  
 plt.legend()  
  
 plt.tight\_layout()  
 plt.show()

Точка входа программы и исполнение основного кода

def main():  
 t\_free\_fall, H\_free\_fall, V\_free\_fall = simulate\_free\_fall(H0, V0y, g\_L, dt)  
  
 engine\_idx, t\_powered, H\_powered, V\_powered, a\_powered = find\_engine\_start\_point(  
 t\_free\_fall, H\_free\_fall, V\_free\_fall, M, m, dt  
 )  
  
 if engine\_idx is not None:  
 t\_total = np.concatenate((t\_free\_fall[:engine\_idx + 1], t\_free\_fall[engine\_idx] + t\_powered[1:]))  
 H\_total = np.concatenate((H\_free\_fall[:engine\_idx + 1], H\_powered[1:]))  
 V\_total = np.concatenate((V\_free\_fall[:engine\_idx + 1], V\_powered[1:]))  
 a\_total = np.concatenate((np.full(engine\_idx + 1, g\_L), a\_powered[:-1]))  
  
 plot\_results(t\_total, H\_total, V\_total, a\_total)  
  
 print(f"Высота на которой нужно включить двигатель: {H\_free\_fall[engine\_idx]:.2f} м")  
 print(f"Вертикальная скорость при посадке: {V\_total[-1]:.2f} м/с")  
 else:  
 print("Безопасная посадка невозможна")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

7. Графики

Изображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

8. Окончательные результаты.

8.1 На основе численного моделирования движения аппарата по вертикальной оси были построены графики зависимости высоты H, вертикальной скорости Vy​ и ускорения ay​ от времени.

8.2 Итоговая вертикальная скорость аппарата при посадке составила примерно 0.6 м/с, двигатель нужно будет включить на высоте примерно 136.1 м.

8.3 Достигнутая вертикальная скорость значительно меньше безопасной посадочной скорость в 3 м/с, что говорит о возможности безопасного приземления.

9. Выводы и анализ результатов работы.

9.1 Построены графики, демонстрирующие изменение высоты, скорости и ускорения аппарата при его спуске, включая моменты свободного падения и работы двигателя.

9.2 Высота для безопасного включения двигателя составляет примерно **136.1 м**. Это позволяет начать торможение на оптимальной высоте для снижения посадочной скорости.

9.3 Вертикальная скорость аппарата при посадке составила около **0.6 м/с**, что значительно ниже допустимых 3 м/с, обеспечивая безопасную посадку.

Модель показала, что при заданных параметрах аппарат может безопасно приземлиться, если включить двигатель на рассчитанной высоте.