**月球软着陆控制系统综合仿真及分析报告**

**1 引言**

**1.1 研究背景**

在月球探测带来巨大利益的驱使下，世界各国纷纷出台了自己的探月计划，

再一次掀起了新一轮探月高潮。在月球上着陆分为两种，一种称为硬着陆，顾名

思义，就是探测器在接近月球时不利用制动发动机减速而直接撞击月球。另一种

称为软着陆，这种着陆方式要求探测器在距月面一定高度时开启制动系统，把探

测器的速度抵消至零，然后利用小推力发动机把探测器对月速度控制在很小的范

围内，从而使其在着陆时的速度具有几米每秒的数量级。显然，对于科学研究，

对探测器实施月球软着陆的科学价值要大于硬着陆。

**1.2 设计目标**

1) 建立月球软着陆制导与控制系统仿真模型并进行仿真。

2) 对相关参数对软着陆性能的影响进行分析，包括以下几个方面：

(1) 发动机推力偏差(±10%)对软着陆过程的影响；

(2) 比冲偏差(±10%)对软着陆过程的影响；

(3) 初始速度方向偏差对软着陆过程的影响；

(4) 其它参数对软着陆过程的影响。

**2 理论基础与模型建立**

**2.1 月球软着陆过程**

本仿真实验以以下软着陆方法为基础：先经过一条绕月停泊轨道，然后再伺机制动下降到月球表面。探测器首先沿着飞月轨道飞向月球，在距月球表面一定高度时，动力系统给探测器施加一制动脉冲，使其进入一条绕月运行的停泊轨道；

然后根据事先选好的着陆点，选择霍曼变轨起始点，给探测器施加一制动脉冲，使其进入一条椭圆形的下降轨道，最后在近月点实施制动减速以实现软着陆。

**2.2 坐标系定义**

首先定义两个月球软着陆坐标系。第一个是月心惯性坐标系：原点

选在月心，轴指向动力下降起始点，轴垂直于轴指向着陆点方向，轴按右手法则确定。

探测器在空间的位置可由表示成球坐标的形式，为从月心到探测器的距离，表示月球经度和纬度。

第二个就是探测器轨道坐标系：原点选在探测器质心，轴与从月心到探测器质心的矢径方向重合，背离月心方向为正，轴垂直于轴指向运动方向为正，按右手法则确定。

**2.3 动力学模型**

假设制动发动机为常推力液体发动机，忽略月球自转，则月球软着陆动力学方程可表示为：

(2-1)

其中：

(2-2)

是探测器月心距矢径， ,分别表示径向加速度和速度，为轨道坐标系相对

惯性系的角速度矢量，为制动推力开关控制函数，为月球引力常数，为探测器质量。用、、表示上述动力学方程可得

(2-3)

其中：为推力发动机推力大小，、为轨道坐标系下推力矢量的方向角；

，为发动机比冲，为地表重力加速度常数。

**2.4 制导律设计**

本实验采用是一种基于多项式显式制导的燃耗次优控制制导律，该制导律由以下公式构成：

1. 径向加速度计算公式

(2-4)

1. 控制变量（推力俯仰角、偏航角）计算公式

(2-5)

1. 剩余时间估计公式

(2-6)

其中，和可由加速度仪实时测得。分析上述公式可以看出，该制导律是剩余时间的函数，而剩余时间只与探测器当前状态和末端约束状态有关。

**3 仿真系统设计与实现**

**3.1 仿真平台架构**

本实验基于MATLAB环境设计了一个闭环仿真框架，主要包含以下模块：

（1）动力学模型（6自由度）

（2）制导律模块（多项式显式制导的燃耗次优控制）

（3）控制器模块（推力方向控制）

（4）偏差注入模块（推力、比冲、初始速度方向等）

（5）分析记录模块（记录终端状态和过程数据并可视化结果）

具体步骤如下：

（1）定义全局参数（具体参数定义程序详见附录A）

（2）建立动力学模型（ODE函数）：在球坐标系下，根据式(2-3)编写微分方程组。

（3）制导律模块（函数形式）：输入当前状态、终端约束以及当前时间，输出推力方向角psi, phi，以及剩余时间t\_go。

（4）主仿真循环（使用ODE求解器）：使用ode45求解动力学微分方程组，但需要控制输入（制导律）在每一步更新。因此，采用“带有事件函数的循环”，即在每个积分步长调用制导律函数更新psi和phi。

（5）偏差注入：完成标称工况下的仿真之后，使用带有偏差的值（推力、比冲、初始速度）分别再次进行仿真。

（6）记录分析：在以上所有仿真结束后，提取过程以及终端状态，记录不同偏差条件下的结果，并绘制轨迹曲线、高度随时间变化、经纬度随时间变化、质量随时间变化等图，并对比分析不同工况下的结果。

**3.2 核心模块实现**

3.2.1 动力学模型模块

编写函数，输入状态变量、时间以及仿真参数，输出状态导数。

为了避免积分计算时出现奇点，需对输入的beta角进行处理使其不严格为0：

if abs(beta) < 1e-5

if beta == 0

beta = 1e-5;

else

beta = sign(beta) \* 1e-5;

end

end

每个积分步长调用制导律函数更新控制角psi和phi（制导律模块详见下一节）：

[psi, phi, ~] = guidanceLaw(state, params);

根据式(2-3)编写微分方程：

dr = u;

dbeta = v / r;

dalpha = w / (r \* sin(beta));

du = (F \* cos(psi) / m) - params.mu\_moon/r^2 + (v^2 + w^2)/r;

dv = (F \* sin(psi) \* cos(phi) / m) - (u\*v)/r + (w^2)/(r\*tan(beta));

dw = (F \* sin(psi) \* sin(phi) / m) - (u\*w)/r - (v\*w)/(r\*tan(beta));

dm = -F / C;

3.2.1 制导律计算模块

编写多项式显式制导律函数，输入状态变量和仿真参数，输出推力方向角psi, phi、剩余时间估计t\_go。

首先根据状态量计算当前水平速度：

delta\_v = params.vf - v;

delta\_w = params.wf - w;

V\_horizontal = sqrt(delta\_v^2 + delta\_w^2);

假设初始假设a\_H = F/m，计算a\_H：

a\_F = F / m;

a\_H\_initial = a\_F;

根据式(2-6)估计剩余时间：

t\_go\_guess = V\_horizontal / a\_H\_initial;

根据式(2-4)和式(2-5)迭代计算径向加速度和psi角，并确定剩余时间：

max\_iter = 10; % 最大迭代次数

tolerance = 1e-5; % 收敛容差

for iter = 1:max\_iter

% 计算径向加速度a

numerator\_a = 6\*(params.rf - r - u\*t\_go\_guess) - 2\*(params.uf - u)\*t\_go\_guess;

a\_radial = numerator\_a / (t\_go\_guess^2);

% 计算psi角

term\_gravity = params.mu\_moon / r^2;

term\_centrifugal = (v^2 + w^2) / r;

numerator\_psi = a\_radial + term\_gravity - term\_centrifugal;

cos\_psi = numerator\_psi / a\_F;

cos\_psi = max(min(cos\_psi, 1), 0); % 限制在有效范围

psi = acos(cos\_psi);

% 更新水平加速度和剩余时间

a\_H = a\_F \* sin(psi);

t\_go\_new = V\_horizontal / a\_H;

% 检查收敛

if abs(t\_go\_new - t\_go\_guess) < tolerance

% disp([iter, "收敛"])

break;

end

t\_go\_guess = t\_go\_new;

end

t\_go = t\_go\_guess;

根据式(2-5)计算phi角：

Vc\_x = params.vf - v; % 纬度方向速度增量

Vc\_y = params.wf - w; % 经度方向速度增量

phi = acos(Vc\_x/sqrt(Vc\_x^2+Vc\_y^2));

3.2.3 偏差注入模块

为了研究相关参数对软着陆性能的影响，在动力学模型和制导律函数中，实际使用的推力F和比冲Isp可能是标称值，也可能是带有偏差的值。因此可以在全局定义偏差因子，例如：

thrust\_factor = 1.1; % 推力增加10%

Isp\_factor = 0.9; % 比冲减少10%

然后在动力学模型和制导律函数中添加：

F\_actual = F\_nominal \* thrust\_factor; % 推力变化

Isp\_actual = Isp\_nominal \* Isp\_factor; % 比冲变化

对于初始速度方向偏差，在初始状态中原本v0=1692, w0=0，如果存在方向偏差，可以将初始速度矢量旋转一个角度（例如在水平面内旋转5度）：

v = 1692 \* cos(deviation\_angle);

w = 1692 \* sin(deviation\_angle);

并用相似的方式添加到动力学模型和制导律函数中。其中deviation\_angle是初始速度方向偏差角（弧度）。

在初始运行时，需要对偏差因子进行初始化，以计算标称工况下的情况：

if ~isfield(params, 'F\_factor'), params.F\_factor = 1.0; end

if ~isfield(params, 'Isp\_factor'), params.Isp\_factor = 1.0; end

if ~isfield(params, 'velocity\_angle'), params.velocity\_angle = 0; end

3.2.4 积分计算模块

在仿真运行程序中使用ode45积分器对动力学方程进行积分：

odefun = @(t,y) lunarDynamics(t, y, params);

[time, state] = ode45(odefun, params.tspan, initial\_state, options);

其中，在options中设置终止条件：

options = odeset('RelTol', params.reltol, 'AbsTol', params.abstol, ...

'Events', @(t,y) landingEvents(t,y,params.rf));

终止事件由终止事件函数定义：

function [value, isterminal, direction] = landingEvents(~, y, rf)

% 达到目标高度 (高度<2km)

value = y(1) - rf; % 目标高度2km

isterminal(1) = 1; % 触发时终止

direction(1) = -1; % 下降穿过阈值时触发

end

**3.3 仿真参数设置**

本实验仿真参数设置如下：

推力器参数：，，地表引力加速度，

；

月球常数：月球引力常数，月球半径；

初始参数：，，，，，

，；

终端参数： （探测器终止于离月面2km的高度），，，。

**4 仿真结果与分析**

**4.1 标称工况分析**

由仿真得出的结果绘制标准工况下的3D着陆轨迹（在直角坐标系中）如下图所示。

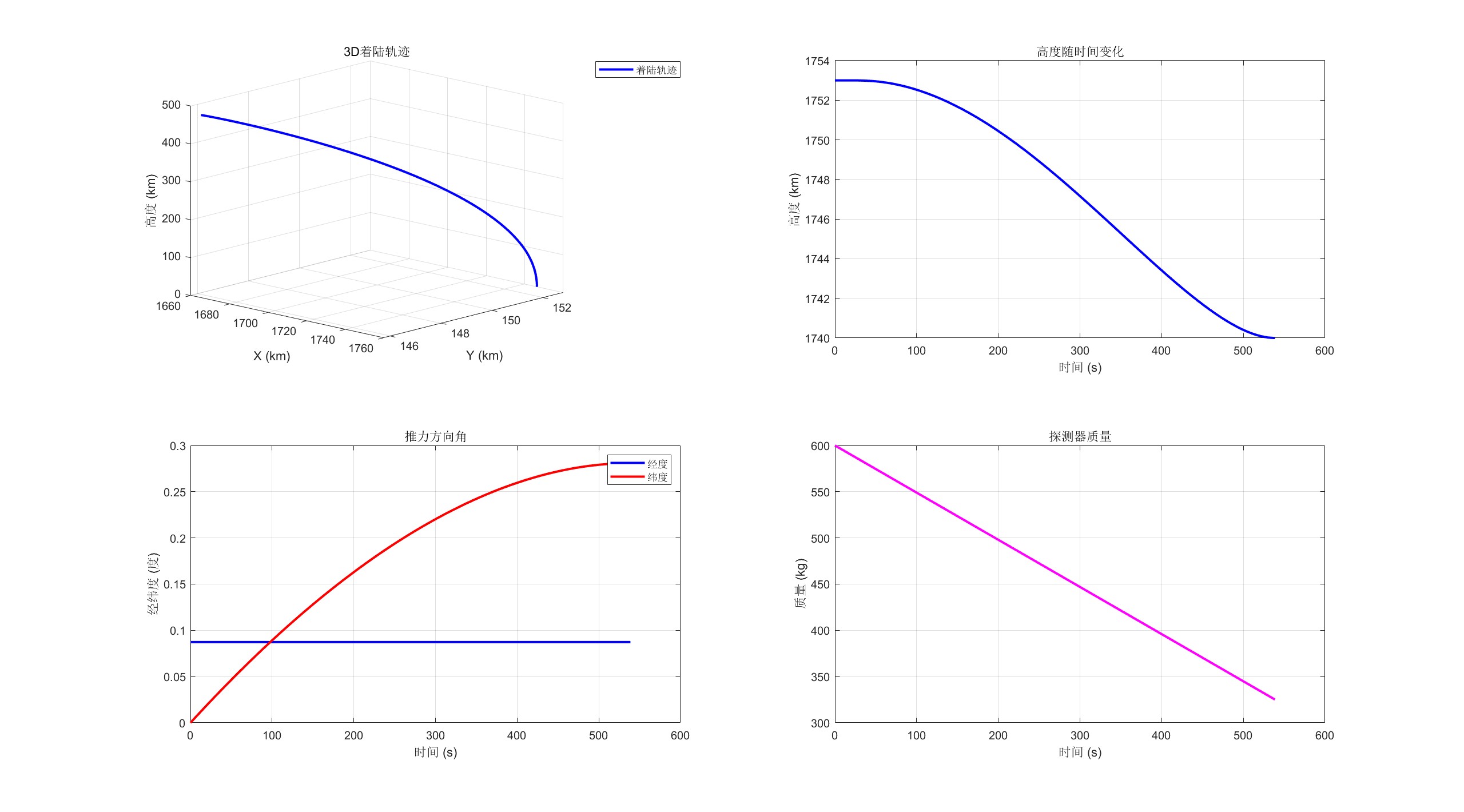


图4-1 标称工况下的3D着陆轨迹

图4-2、图4-3、图4-4分别是标准工况下探测器的高度、经纬度、质量随时间的变化曲线。可以看出，标准工况下的探测器软着陆仿真情况基本符合预期，降落过程耗时约539s，消耗燃料约275kg，燃料消耗与时间呈线性关系。这也验证了多项式显示制导方案的有效性。

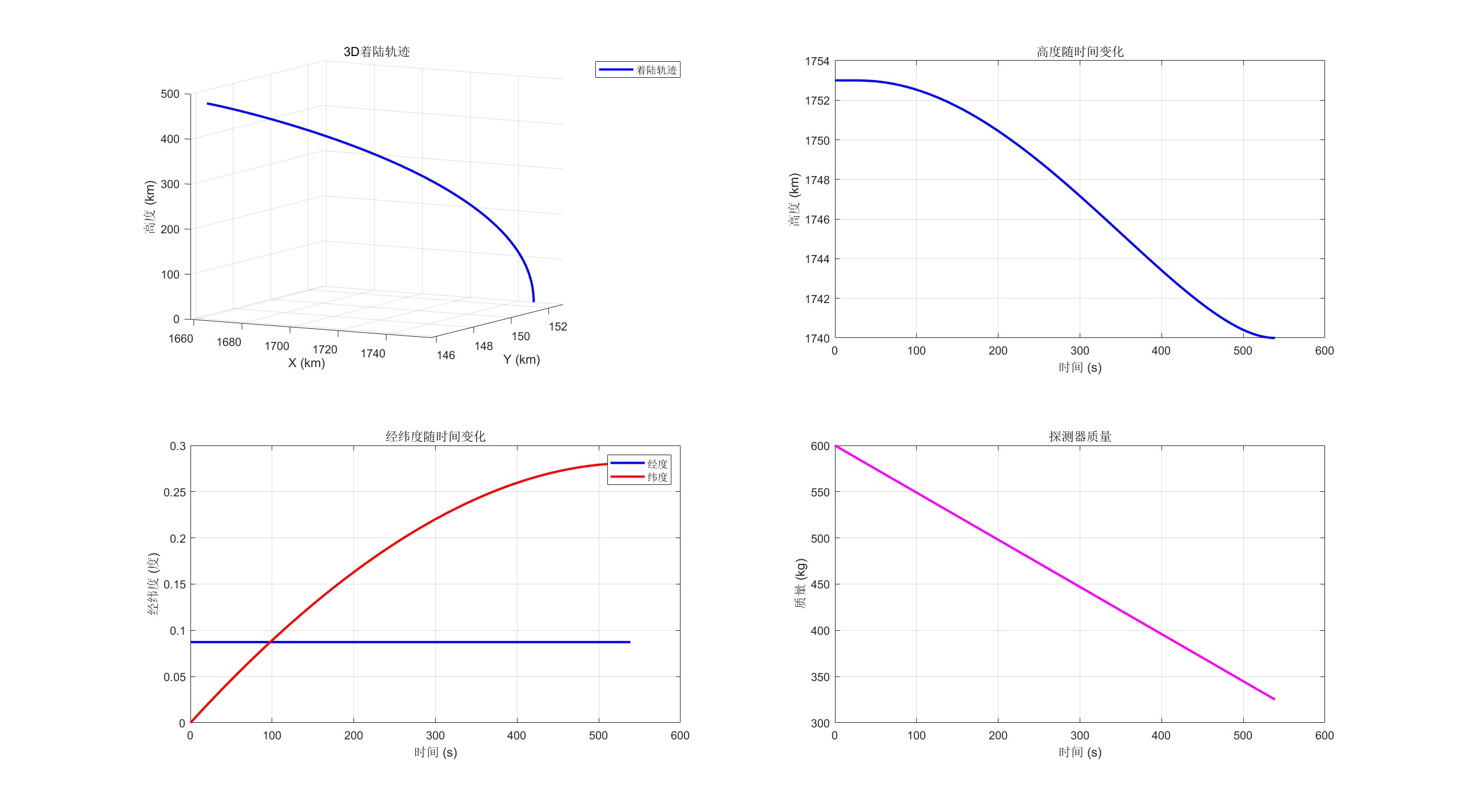


图4-2 标称工况下的探测器高度变化

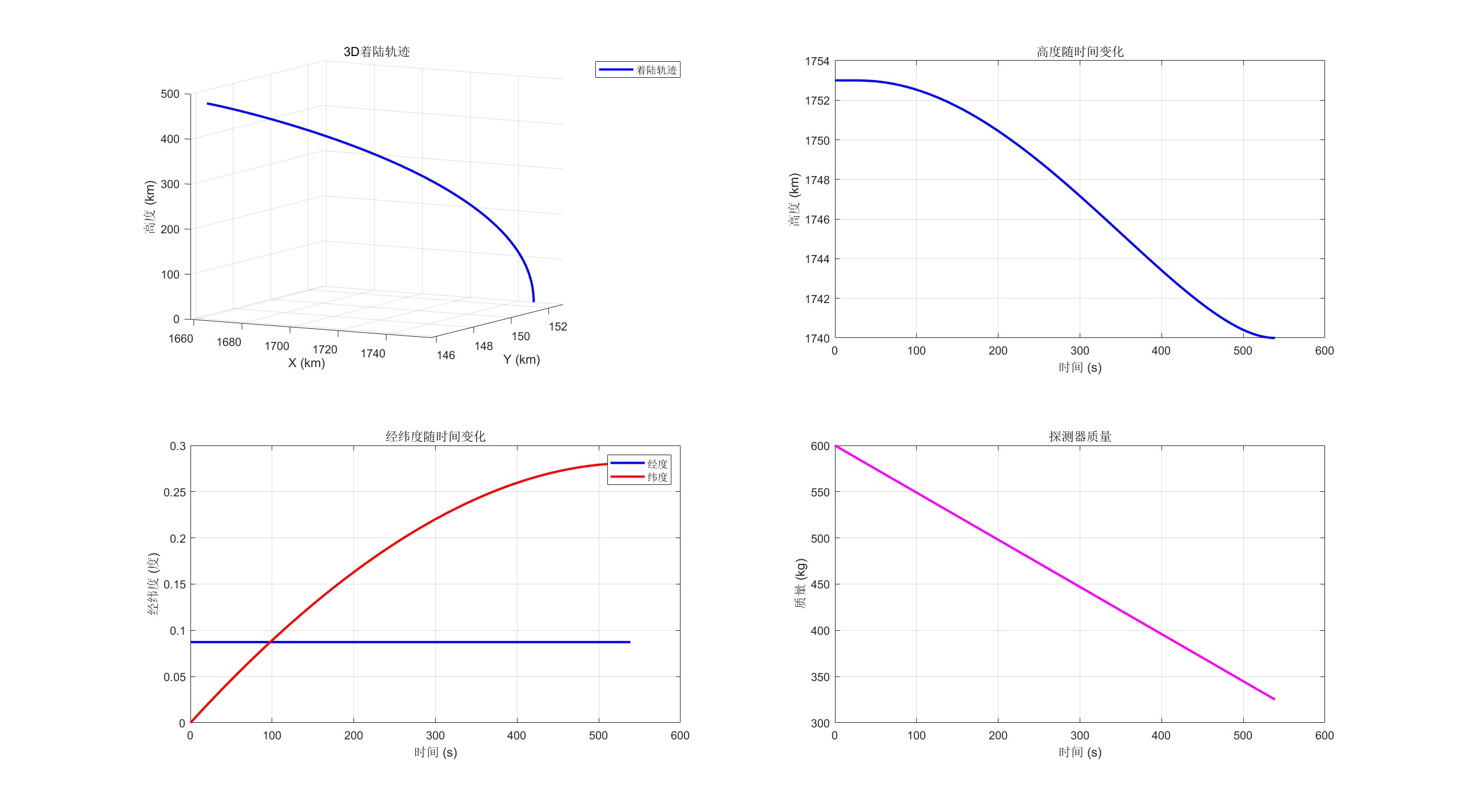


图4-3 标称工况下的探测器经纬度变化

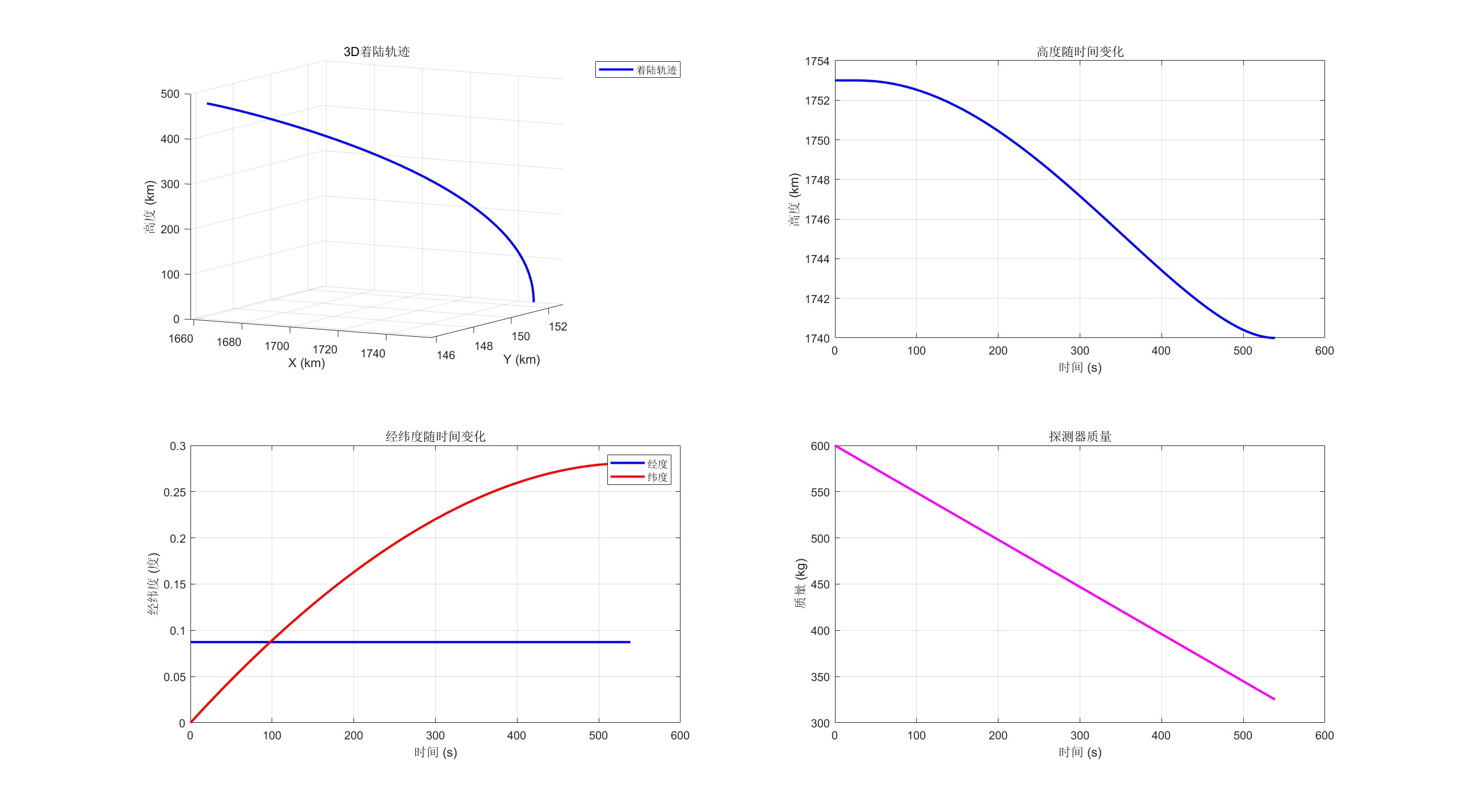


图4-4 标称工况下的探测器质量变化

**4.2 参数偏差影响分析**

4.2.1 推力偏差影响(±10%)

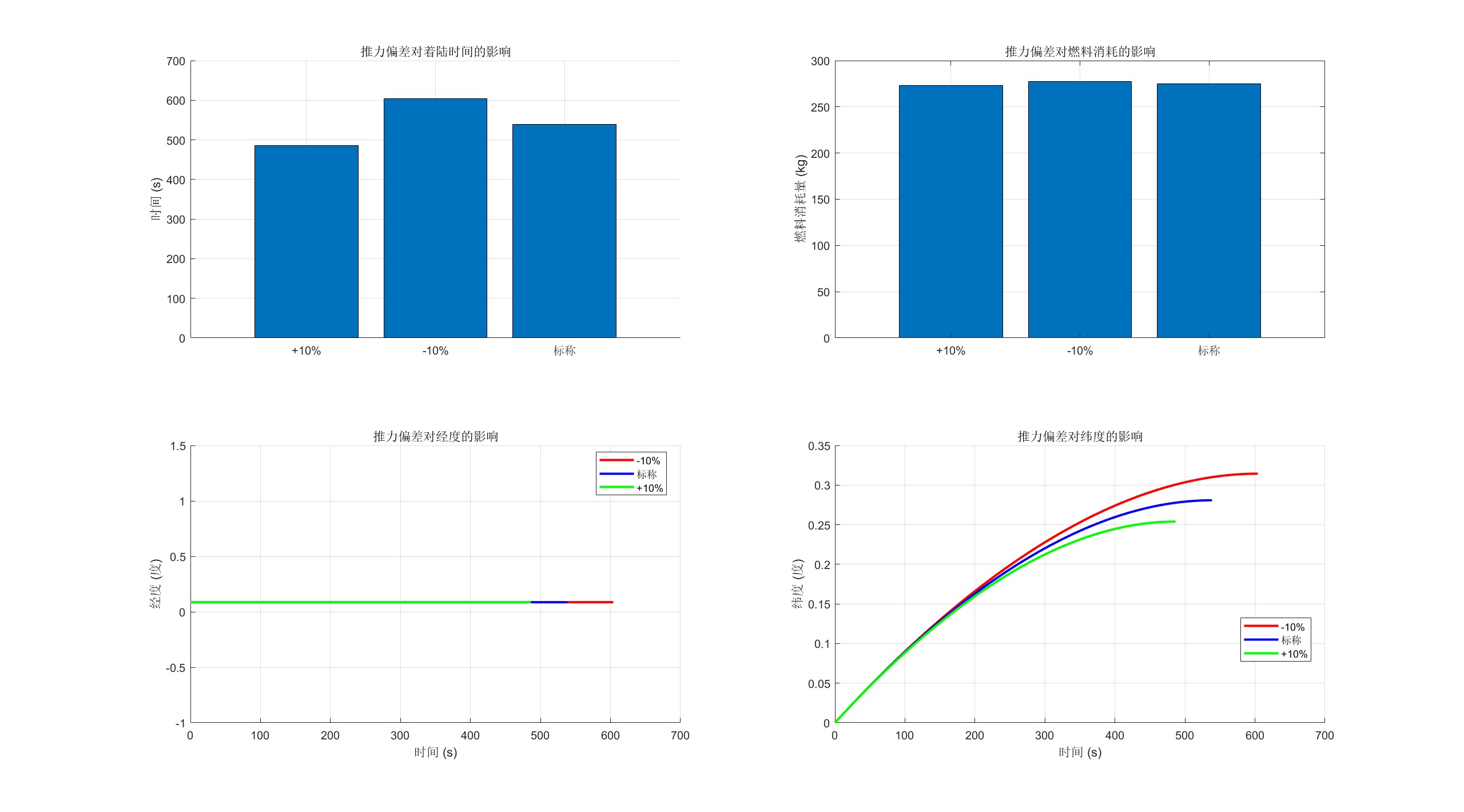


图4-5 推力偏差对着陆时间和燃料消耗的影响

探测器发动机推力偏差（±10%）对着陆时间和燃料消耗的影响如图4-5，对探测器着陆位置（着陆过程中的经纬度）的影响如图4-6所示。

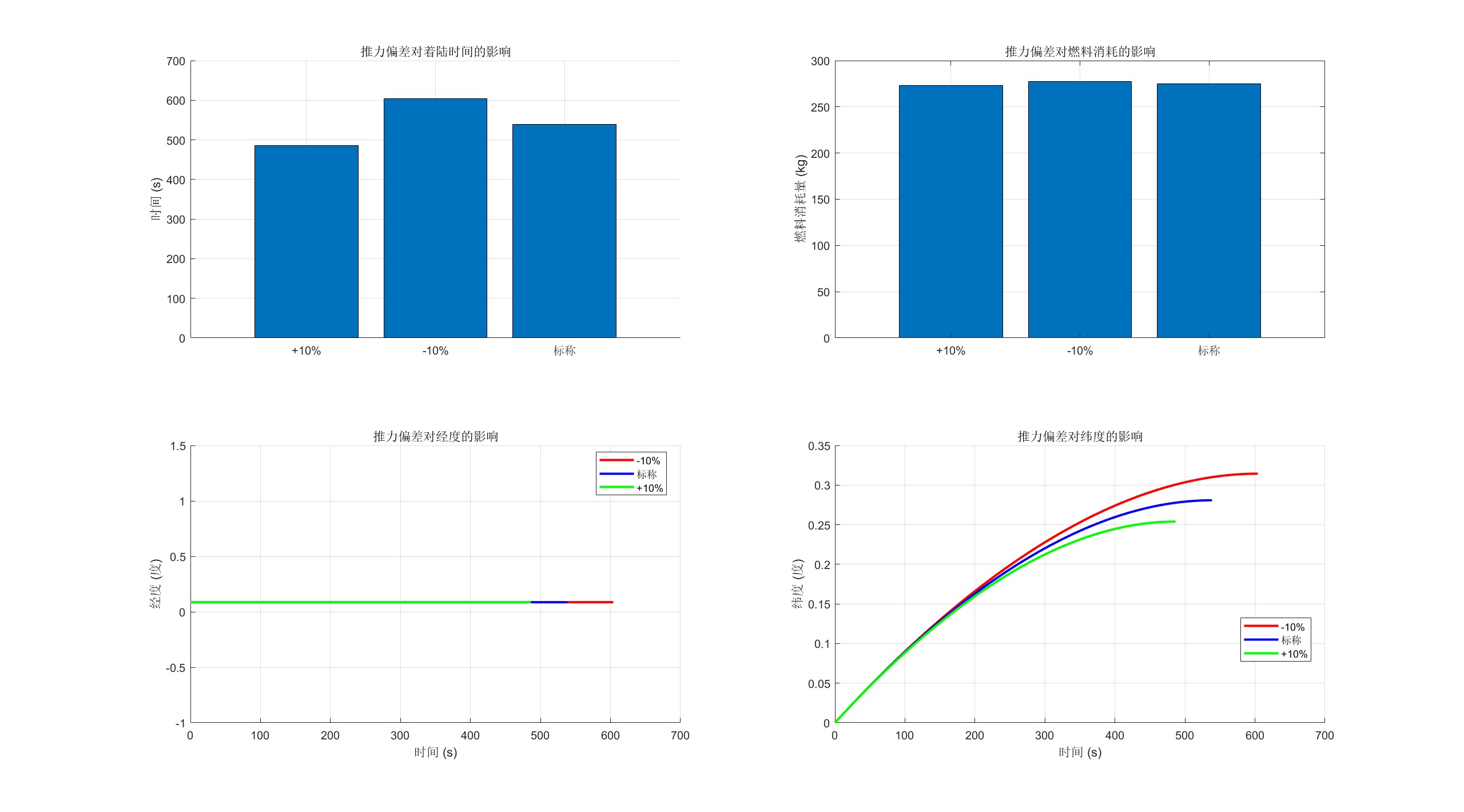


图4-6 推力偏差对探测器经纬度的影响

可以看出，发动机推力偏差对着陆时间影响显著，发动机推力增加10%会使着陆时间缩短约9.7%，推力减少10%会导致着陆时间延长约12.1%，而对燃料消耗影响不大；另外，发动机推力偏差也会影响探测器着陆轨迹纬度变化，推力偏差±10%对探测器终端纬度影响分别约为-12%和+10%，而对经度变化无影响，即终端位置与标况偏差约900m。

4.2.2 比冲偏差影响(±10%)

探测器发动机比冲偏差（±10%）对着陆时间和燃料消耗的影响如图4-7，对探测器着陆位置（着陆过程中的经纬度）的影响如图4-8所示。

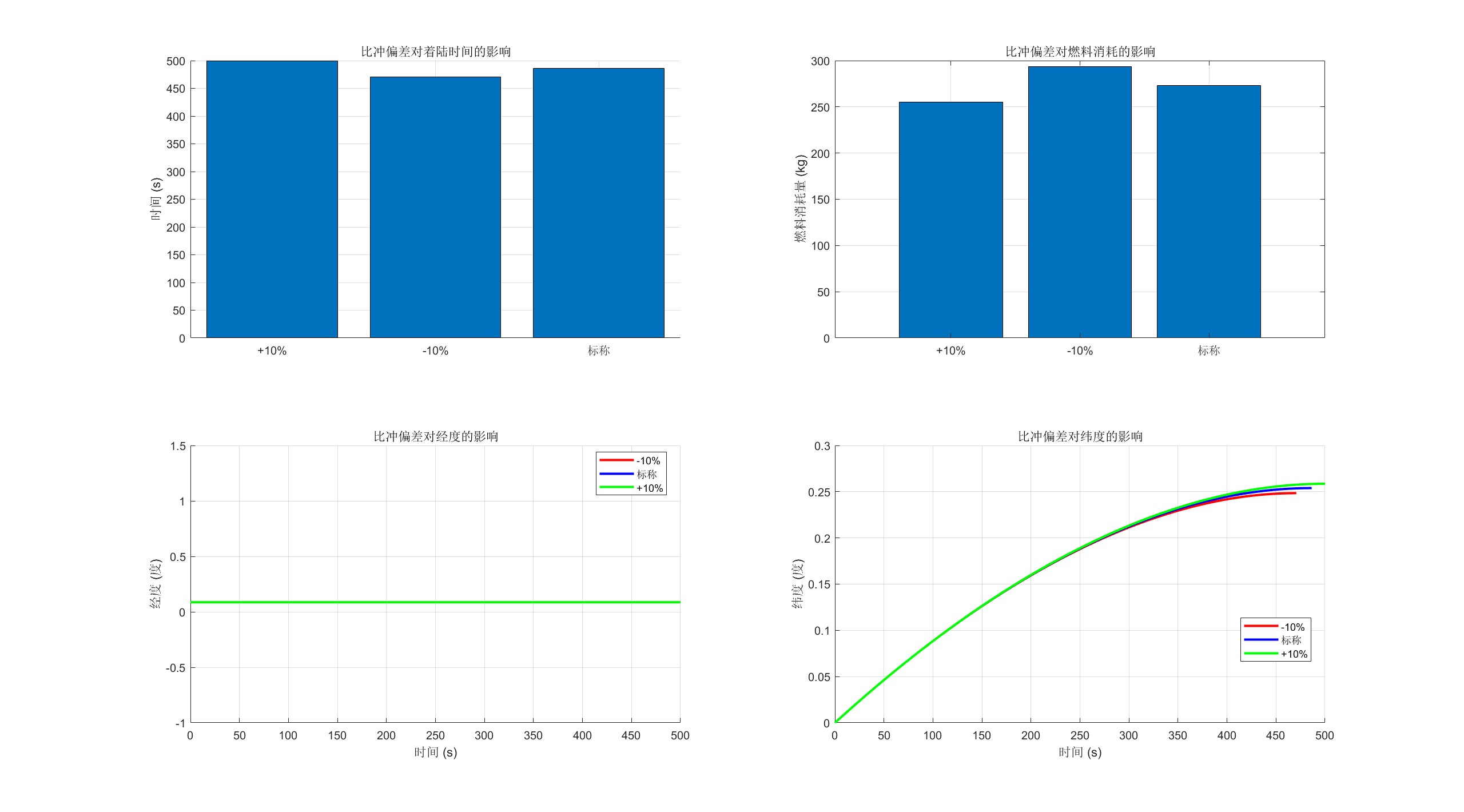


图4-7 比冲偏差对着陆时间和燃料消耗的影响

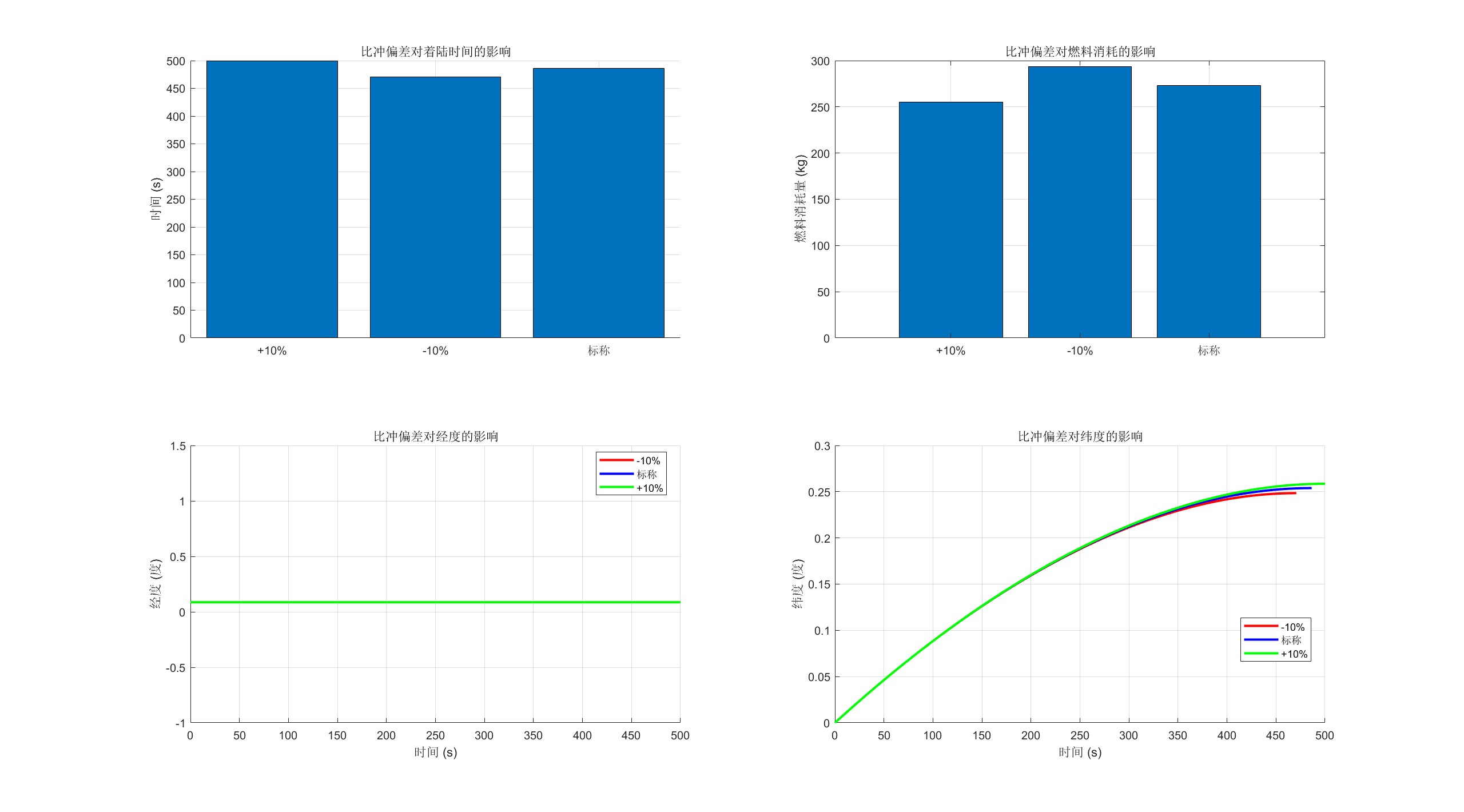


图4-8 比冲偏差对探测器经纬度的影响

可以看出，发动机比冲偏差对着陆时间和燃料消耗均有影响，但对燃料消耗影响更明显。发动机比冲增加10%会使着陆时间延长约2.8%，但燃料消耗减少约6.6%；比冲减少10%会使着陆时间缩短约3.2%，而燃料消耗增加约7.5%。

发动机比冲偏差对探测器着陆轨迹经纬度影响与推力偏差类似，对经度无影响，但对纬度影响更小。比冲偏差±10%对探测器终端纬度影响分别约为+1.8%和-2.2%，即终端位置与标况偏差约150m。

4.2.3 初始速度方向偏差影响

探测器初始速度偏差（水平方向±5°、±3°）对着陆时间和燃料消耗的影响如图4-9，对探测器着陆位置（着陆过程中的经纬度）的影响如图4-10所示。

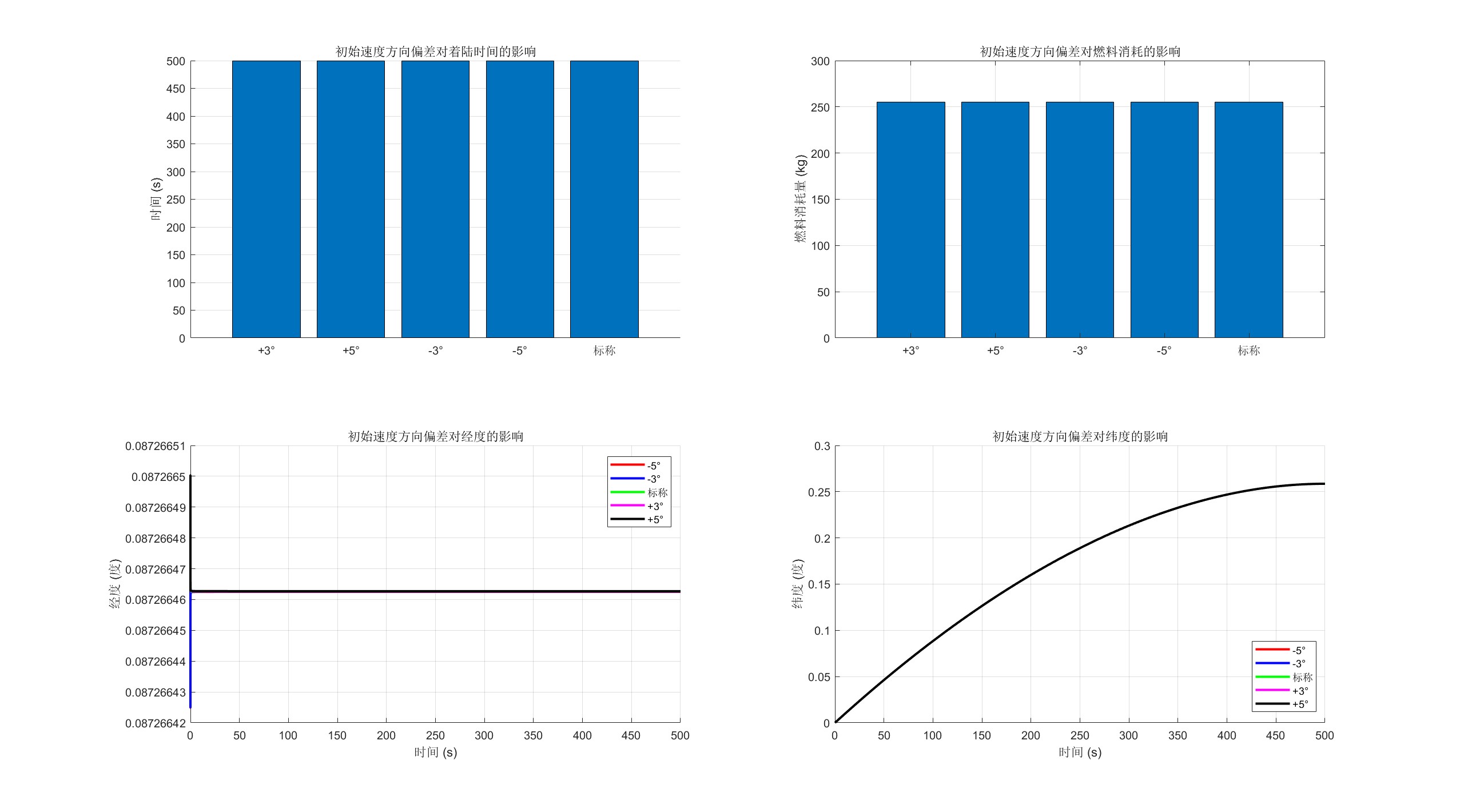


图4-9 初始速度偏差对着陆时间和燃料消耗的影响

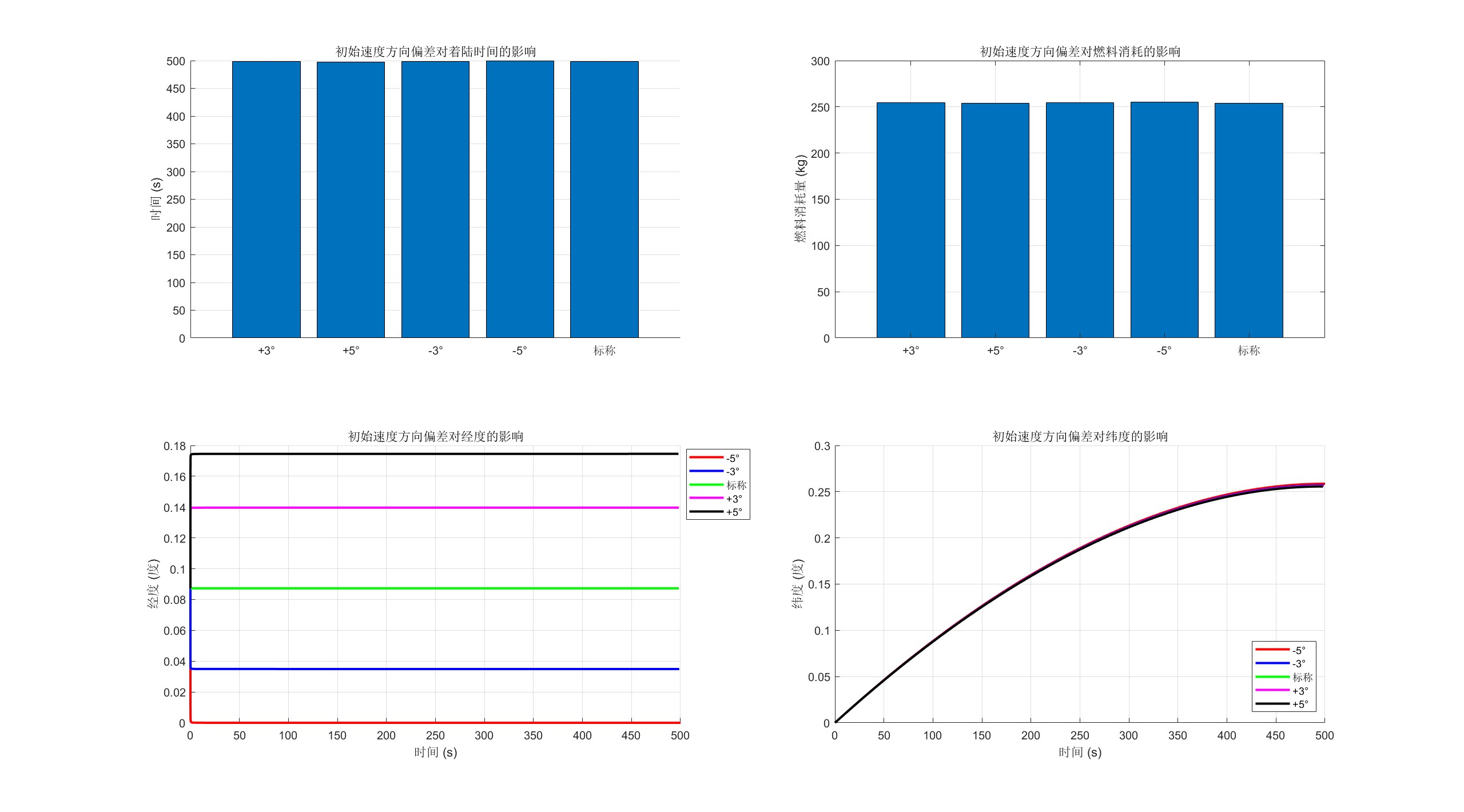


图4-10 初始速度偏差对探测器经纬度的影响

可以看出，探测器初始速度方向偏差对着陆时间和燃料消耗无影响，对探测器纬度变化影响很小，但对经度影响较为明显。探测器初始水平速度偏差±3°时，探测器经度变化了±60%，终端位置偏差约1.6km；探测器初始水平速度偏差±5°时，探测器经度变化了±100%，终端位置偏差约2.7km。

**5 结论与展望**

**5.1 主要结论**

1. 制导律性能：标准工况下的探测器软着陆仿真情况基本符合预期，降落过程耗时约539s，消耗燃料约275kg，验证了多项式显示制导方案的有效性。

2. 参数敏感度：

（1）推力偏差对着陆时间影响显著，发动机推力增加10%会使着陆时间缩短约9.7%，推力减少10%会导致着陆时间延长约12.1%，；影响探测器着陆轨迹纬度变化，推力偏差±10%对探测器终端纬度影响约为10%，终端位置偏差约＜1km。

（2）比冲偏差对燃料消耗影响显著，发动机比冲增加10%会使燃料消耗减少约6.6%，比冲减少10%会使燃料消耗增加约7.5%。比冲偏差±10%对探测器终端纬度影响约为2%，终端位置偏差＜200m。

（3）初始速度方向偏差对经度影响显著，探测器初始水平速度偏差±3°时，探测器经度变化了±60%，终端位置偏差＞1km；探测器初始水平速度偏差±5°时，探测器经度变化了±100%，终端位置偏差＞2km。

**5.2 不足与展望**

（1）本仿真实验所使用的模型未考虑月球自传等复杂因素，具有一定的局限性，可在模型设计上考虑更多实际因素并提高模型经度；

（2）本仿真实验所使用的制导律对一些参数敏感，发动机推力和比冲的偏差会影响着陆时间和燃料消耗，初始速度偏差可能会导致着陆点大幅偏移，可以通过在制导律设计中引入偏差补偿项等方法提高制导律鲁棒性；

（3）本仿真实验所使用的制导律多项式近似模型忽略高阶项，导致末端轨迹精度下降，可以通过在加速度公式中增加终端收敛项，确保末端速度精确收敛至零，避免突变。

**附录**

**A. 完整程序代码**

主程序 LunarSoftLandingSim.m

%% 主仿真脚本

% 月球软着陆制导控制系统综合仿真

clearvars; close all; clc;

%% 1. 系统参数初始化

simParams = struct();

% 物理常数

simParams.gE = 9.8; % 地表重力加速度 [m/s²]

simParams.mu\_moon = 4.88775e12; % 月球引力常数 [m³/s²]

simParams.R\_moon = 1738e3; % 月球半径 [m]

% 推进系统参数

simParams.F\_nominal = 1500; % 标称推力 [N]

simParams.Isp\_nominal = 300; % 标称比冲 [s]

simParams.C\_nominal = simParams.Isp\_nominal \* simParams.gE; % 发动机常数

% 初始状态

simParams.r0 = 1753e3; % 初始月心距 [m]

simParams.beta0 = deg2rad(0); % 初始纬度 [rad]

simParams.alpha0 = deg2rad(5); % 初始经度 [rad]

simParams.u0 = 0; % 初始径向速度 [m/s]

simParams.v0 = 1692; % 初始横向速度1 [m/s]

simParams.w0 = 0; % 初始横向速度2 [m/s]

simParams.m0 = 600; % 初始质量 [kg]

% 终端约束

simParams.rf = 1740e3; % 终端月心距 [m] (距月面2km)

simParams.uf = 0; % 终端径向速度 [m/s]

simParams.vf = 0; % 终端横向速度1 [m/s]

simParams.wf = 0; % 终端横向速度2 [m/s]

% 仿真设置

simParams.tspan = [0, 2000]; % 仿真时间 [s]

simParams.reltol = 1e-6; % 相对容差

simParams.abstol = 1e-6; % 绝对容差

% 事件函数：检测高度是否达到终端

options = odeset('Events', @(t,y) landingEvents(t, y, simParams.rf));

%% 2. 运行标称工况仿真

[time\_nominal, state\_nominal] = runLandingSimulation(simParams);

%% 3. 参数偏差影响分析

% 3.1 推力偏差分析 (±10%)

thrust\_deviations = [0.9, 1.0, 1.1]; % -10%, 标称, +10%

time\_thrust = cell(size(thrust\_deviations));

state\_thrust = cell(size(thrust\_deviations));

for i = 1:length(thrust\_deviations)

simParams.F\_factor = thrust\_deviations(i);

[time\_thrust{i}, state\_thrust{i}] = runLandingSimulation(simParams);

end

% 3.2 比冲偏差分析 (±10%)

Isp\_deviations = [0.9, 1.0, 1.1]; % -10%, 标称, +10%

time\_Isp = cell(size(Isp\_deviations));

state\_Isp = cell(size(Isp\_deviations));

for i = 1:length(Isp\_deviations)

simParams.Isp\_factor = Isp\_deviations(i);

[time\_Isp{i}, state\_Isp{i}] = runLandingSimulation(simParams);

end

% 3.3 初始速度方向偏差分析

velocity\_angles = deg2rad([-5, -3, 0, 3, 5]); % 偏差角度 [rad]

time\_velocity = cell(size(velocity\_angles));

state\_velocity = cell(size(velocity\_angles));

for i = 1:length(velocity\_angles)

simParams.velocity\_angle = velocity\_angles(i);

% 处理初始速度方向偏差

v\_corrected = simParams.v0 \* cos(simParams.velocity\_angle);

w\_corrected = simParams.v0 \* sin(simParams.velocity\_angle);

simParams.v0 = v\_corrected;

simParams.w0 = w\_corrected;

[time\_velocity{i}, state\_velocity{i}] = runLandingSimulation(simParams);

end

%% 4. 结果可视化与分析

analyzeResults(time\_nominal, state\_nominal, time\_thrust, state\_thrust, time\_Isp, state\_Isp, time\_velocity, state\_velocity);

核心功能模块：

动力学模型 lunarDynamics.m

function dstate = lunarDynamics(t, state, params)

% 月球软着陆6自由度动力学模型

% 解包状态变量

r = state(1); % 月心距 [m]

beta = state(2); % 纬度 [rad]

u = state(4); % 径向速度 [m/s]

v = state(5); % 纬度方向速度 [m/s]

w = state(6); % 经度方向速度 [m/s]

m = state(7); % 质量 [kg]

% 避免奇点 (beta = 0)

if abs(beta) < 1e-5

if beta == 0

beta = 1e-5;

else

beta = sign(beta) \* 1e-5;

end

end

% 调用制导律计算控制角

[psi, phi] = guidanceLaw(state, params);

% 应用参数偏差

F = params.F\_nominal \* params.F\_factor;

Isp = params.Isp\_nominal \* params.Isp\_factor;

C = Isp \* params.gE;

% 动力学方程 (式17-2)

dr = u;

dbeta = v / r;

dalpha = w / (r \* sin(beta));

du = (F \* cos(psi) / m) - params.mu\_moon/r^2 + (v^2 + w^2)/r; % 径向加速度

dv = (F \* sin(psi) \* cos(phi) / m) - (u\*v)/r + (w^2)/(r\*tan(beta)); % 横向加速度

dw = (F \* sin(psi) \* sin(phi) / m) - (u\*w)/r - (v\*w)/(r\*tan(beta));

dm = -F / C; % 质量变化

% 返回状态导数

dstate = [dr; dbeta; dalpha; du; dv; dw; dm];

end

制导律模块 guidanceLaw.m

function [psi, phi, t\_go] = guidanceLaw(state, params)

% 多项式显式制导律

% 输入: 当前状态, 系统参数

% 输出: 推力方向角(psi, phi), 剩余时间估计(t\_go)

% 解包状态变量

r = state(1);

u = state(4);

v = state(5);

w = state(6);

m = state(7);

delta\_v = params.vf - v;

delta\_w = params.wf - w;

V\_horizontal = sqrt(delta\_v^2 + delta\_w^2); % 当前水平速度

% 初始猜测：假设a\_H = F/m

F = params.F\_nominal \* params.F\_factor;

a\_F = F / m;

a\_H\_initial = a\_F;

% 剩余时间估计 (式17-16)

t\_go\_guess = V\_horizontal / a\_H\_initial;

max\_iter = 10; % 最大迭代次数

tolerance = 1e-5; % 收敛容差

for iter = 1:max\_iter

% 计算径向加速度a（式17-14）

numerator\_a = 6\*(params.rf - r - u\*t\_go\_guess) - 2\*(params.uf - u)\*t\_go\_guess;

a\_radial = numerator\_a / (t\_go\_guess^2);

% 计算psi角（式17-15）

term\_gravity = params.mu\_moon / r^2;

term\_centrifugal = (v^2 + w^2) / r;

numerator\_psi = a\_radial + term\_gravity - term\_centrifugal;

cos\_psi = numerator\_psi / a\_F;

cos\_psi = max(min(cos\_psi, 1), 0); % 限制在有效范围

psi = acos(cos\_psi);

% 更新水平加速度和剩余时间

a\_H = a\_F \* sin(psi);

t\_go\_new = V\_horizontal / a\_H;

% 检查收敛

if abs(t\_go\_new - t\_go\_guess) < tolerance

% disp([iter, "收敛"])

break;

end

t\_go\_guess = t\_go\_new;

end

t\_go = t\_go\_guess;

% 推力偏航角phi计算

Vc\_x = params.vf - v; % 纬度方向速度增量

Vc\_y = params.wf - w; % 经度方向速度增量

phi = acos(Vc\_x/sqrt(Vc\_x^2+Vc\_y^2));

end

仿真运行函数 runLandingSimulation.m

function [time, state] = ...

runLandingSimulation(params)

% 执行单次着陆仿真

% 初始化控制参数

if ~isfield(params, 'F\_factor'), params.F\_factor = 1.0; end

if ~isfield(params, 'Isp\_factor'), params.Isp\_factor = 1.0; end

if ~isfield(params, 'velocity\_angle'), params.velocity\_angle = 0; end

% 初始状态向量

initial\_state = [params.r0; params.beta0; params.alpha0; ...

params.u0; params.v0; params.w0; params.m0];

% 定义事件函数 (终止条件)

options = odeset('RelTol', params.reltol, 'AbsTol', params.abstol, ...

'Events', @(t,y) landingEvents(t,y,params.rf));

% 使用ode45求解器

odefun = @(t,y) lunarDynamics(t, y, params);

[time, state] = ...

ode45(odefun, params.tspan, initial\_state, options);

end

结果分析绘图函数 analyzeResults.m

function analyzeResults(t1, s1, t2, s2, t3, s3, t4, s4)

% 可视化仿真结果

colors = {'r', 'b', 'g', 'm', 'k'};

%% 1. 标称工况轨迹

figure('Name', '标称工况着陆轨迹');

% 3D轨迹

subplot(2,2,1);

[x, y, z] = sph2cart(s1(:,3), s1(:,2), s1(:,1));

plot3(x/1000, y/1000, z/1000, 'b', 'LineWidth', 2);

title('3D着陆轨迹');

xlabel('X (km)'); ylabel('Y (km)'); zlabel('高度 (km)');

grid on; legend('着陆轨迹');

% 高度变化

subplot(2,2,2);

plot(t1, s1(:,1)/1000, 'b', 'LineWidth', 2);

title('高度随时间变化');

xlabel('时间 (s)'); ylabel('高度 (km)');

grid on;

% 经纬度变化

subplot(2,2,3);

plot(t1, s1(:,3), 'b-', 'LineWidth', 2);

hold on;

plot(t1, s1(:,2), 'r-', 'LineWidth', 2);

title('推力方向角');

xlabel('时间 (s)'); ylabel('经纬度 (度)');

legend('经度', '纬度');

grid on;

% 质量变化

subplot(2,2,4);

plot(t1, s1(:,7), 'm-', 'LineWidth', 2);

title('探测器质量');

xlabel('时间 (s)'); ylabel('质量 (kg)');

grid on;

%% 2. 推力偏差影响分析

figure('Name', '推力偏差影响分析');

thrust\_labels = {'-10%', '标称', '+10%'};

subplot(2,2,1);

hold on;

time\_thrust = zeros(1,3);

for i = 1:3

time\_thrust(i) = t2{i}(end);

end

bar(categorical(thrust\_labels), time\_thrust);

title('推力偏差对着陆时间的影响');

ylabel('时间 (s)');

grid on;

subplot(2,2,2);

fuel\_consumed\_thrust = zeros(1,3);

for i = 1:3

fuel\_consumed\_thrust(i) = 600 - s2{i}(end, 7);

end

bar(categorical(thrust\_labels), fuel\_consumed\_thrust);

title('推力偏差对燃料消耗的影响');

ylabel('燃料消耗量 (kg)');

grid on;

subplot(2,2,3);

hold on;

for i = 1:3

plot(t2{i}, s2{i}(:,3), '-', 'Color', colors{i}, 'LineWidth', 2);

end

title('推力偏差对经度的影响');

xlabel('时间 (s)'); ylabel('经度 (度)');

legend({'-10%', '标称', '+10%'}, 'Location', 'best');

grid on;

subplot(2,2,4);

hold on;

for i = 1:3

plot(t2{i}, s2{i}(:,2), '-', 'Color', colors{i}, 'LineWidth', 2);

end

title('推力偏差对纬度的影响');

xlabel('时间 (s)'); ylabel('纬度 (度)');

legend({'-10%', '标称', '+10%'}, 'Location', 'best');

grid on;

%% 3. 比冲偏差影响分析

figure('Name', '比冲偏差影响分析');

Isp\_labels = {'-10%', '标称', '+10%'};

subplot(2,2,1);

hold on;

time\_Isp = zeros(1,3);

for i = 1:3

time\_Isp(i) = t3{i}(end);

end

bar(categorical(thrust\_labels), time\_Isp);

title('比冲偏差对着陆时间的影响');

ylabel('时间 (s)');

grid on;

subplot(2,2,2);

fuel\_consumed\_Isp = zeros(1,3);

for i = 1:3

fuel\_consumed\_Isp(i) = 600 - s3{i}(end, 7);

end

bar(categorical(Isp\_labels), fuel\_consumed\_Isp);

title('比冲偏差对燃料消耗的影响');

ylabel('燃料消耗量 (kg)');

grid on;

subplot(2,2,3);

hold on;

for i = 1:3

plot(t3{i}, s3{i}(:,3), '-', 'Color', colors{i}, 'LineWidth', 2);

end

title('比冲偏差对经度的影响');

xlabel('时间 (s)'); ylabel('经度 (度)');

legend({'-10%', '标称', '+10%'}, 'Location', 'best');

grid on;

subplot(2,2,4);

hold on;

for i = 1:3

plot(t3{i}, s3{i}(:,2), '-', 'Color', colors{i}, 'LineWidth', 2);

end

title('比冲偏差对纬度的影响');

xlabel('时间 (s)'); ylabel('纬度 (度)');

legend({'-10%', '标称', '+10%'}, 'Location', 'best');

grid on;

%% 4. 初始速度方向偏差影响

figure('Name', '初始速度方向偏差影响');

velocity\_labels = {'-5°', '-3°', '标称', '+3°', '+5°'};

subplot(2,2,1);

hold on;

time\_velocity = zeros(1,5);

for i = 1:5

time\_velocity(i) = t4{i}(end);

end

bar(categorical(velocity\_labels), time\_velocity);

title('初始速度方向偏差对着陆时间的影响');

ylabel('时间 (s)');

grid on;

subplot(2,2,2);

fuel\_consumed\_velocity = zeros(1,5);

for i = 1:5

fuel\_consumed\_velocity(i) = 600 - s4{i}(end, 7);

end

bar(categorical(velocity\_labels), fuel\_consumed\_velocity);

title('初始速度方向偏差对燃料消耗的影响');

ylabel('燃料消耗量 (kg)');

grid on;

subplot(2,2,3);

hold on;

for i = 1:5

plot(t4{i}, s4{i}(:,3), '-', 'Color', colors{i}, 'LineWidth', 2);

end

title('初始速度方向偏差对经度的影响');

xlabel('时间 (s)'); ylabel('经度 (度)');

legend({'-5°', '-3°', '标称', '+3°', '+5°'}, 'Location', 'best');

grid on;

subplot(2,2,4);

hold on;

for i = 1:5

plot(t4{i}, s4{i}(:,2), '-', 'Color', colors{i}, 'LineWidth', 2);

end

title('初始速度方向偏差对纬度的影响');

xlabel('时间 (s)'); ylabel('纬度 (度)');

legend({'-5°', '-3°', '标称', '+3°', '+5°'}, 'Location', 'best');

grid on;

end

事件检测函数 landingEvents.m

function [value, isterminal, direction] = landingEvents(~, y, rf)

% 定义仿真终止事件

% 达到目标高度 (高度<2km)

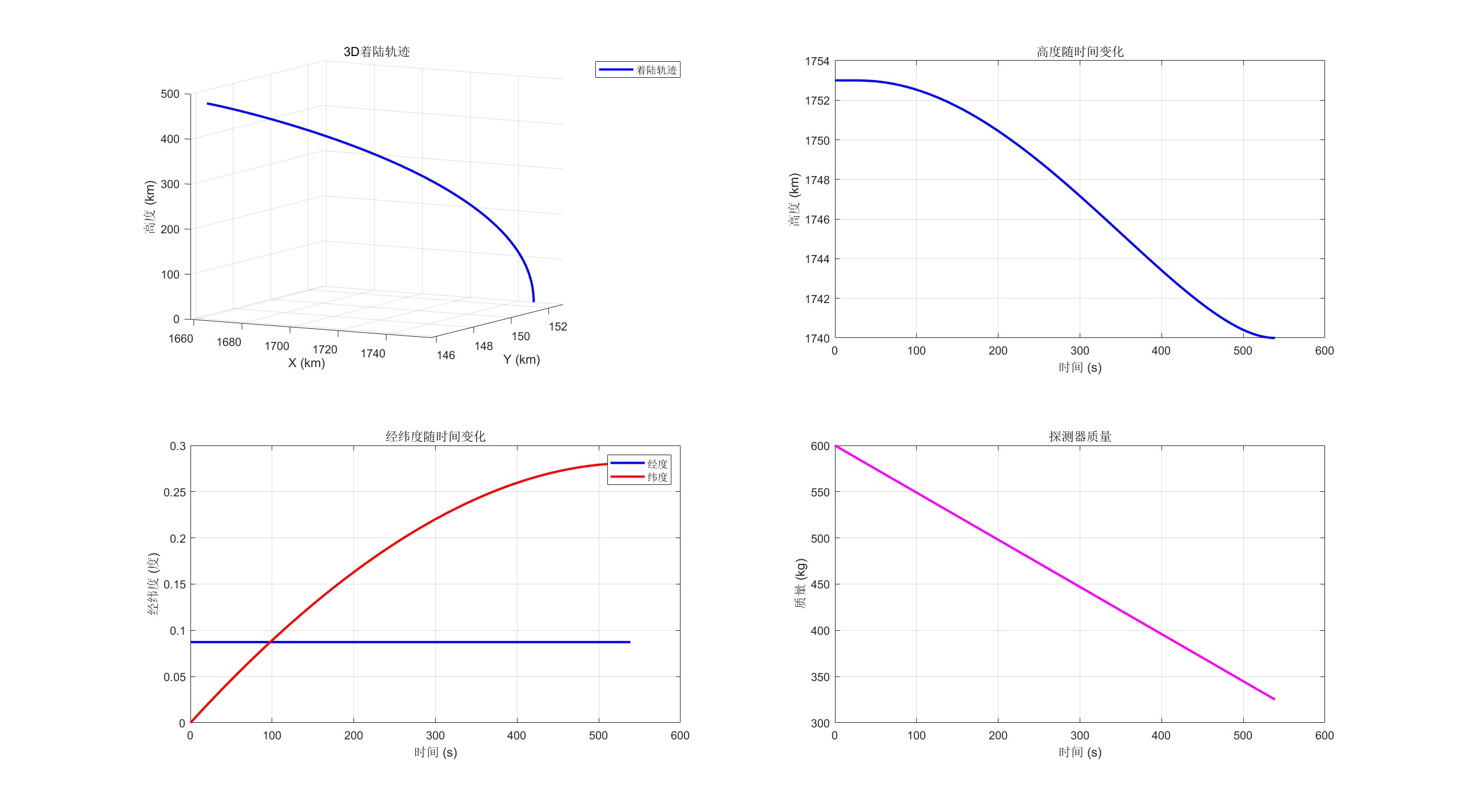
value = y(1) - rf; % 目标高度2km

isterminal(1) = 1; % 触发时终止

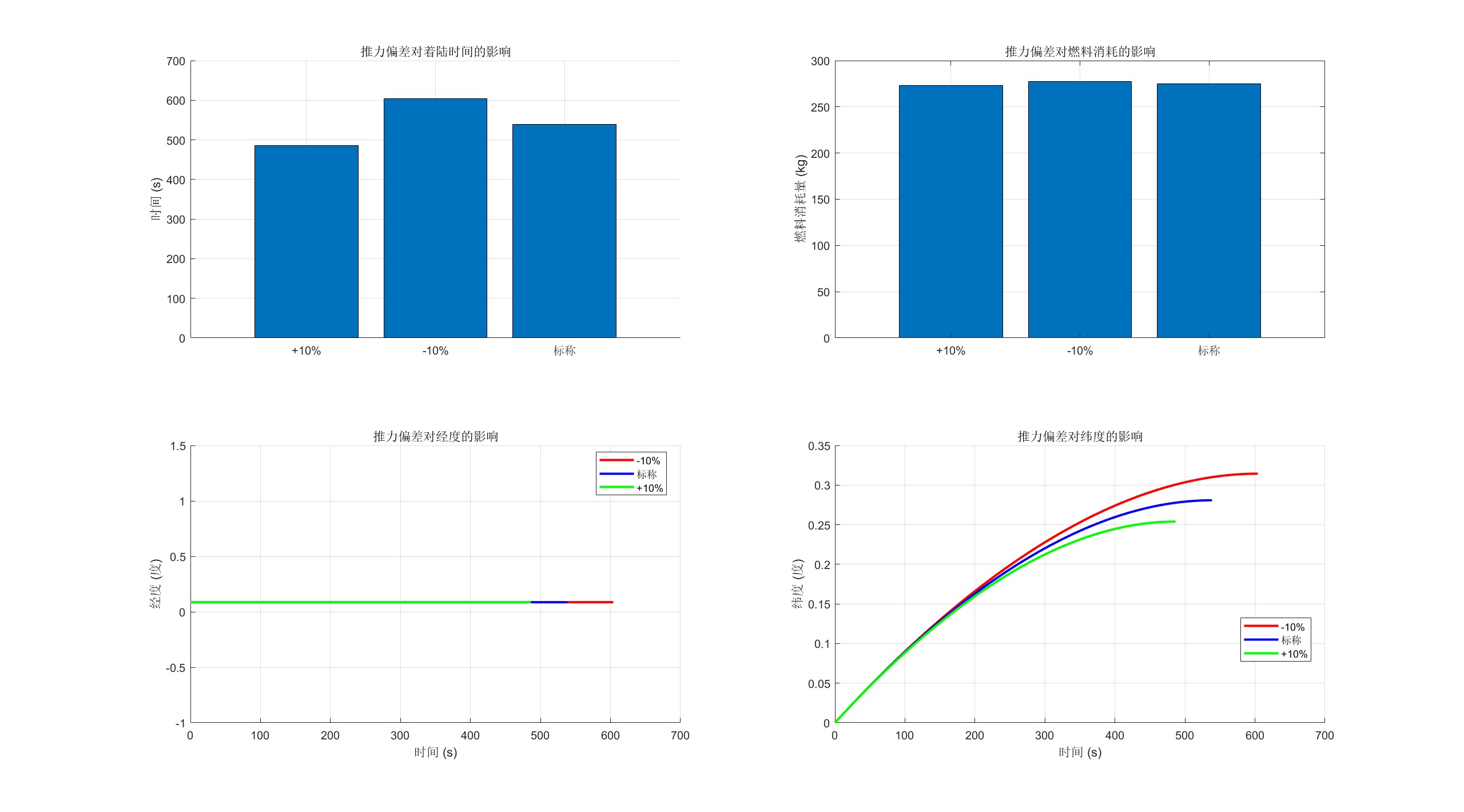
direction(1) = -1; % 下降穿过阈值时触发

end

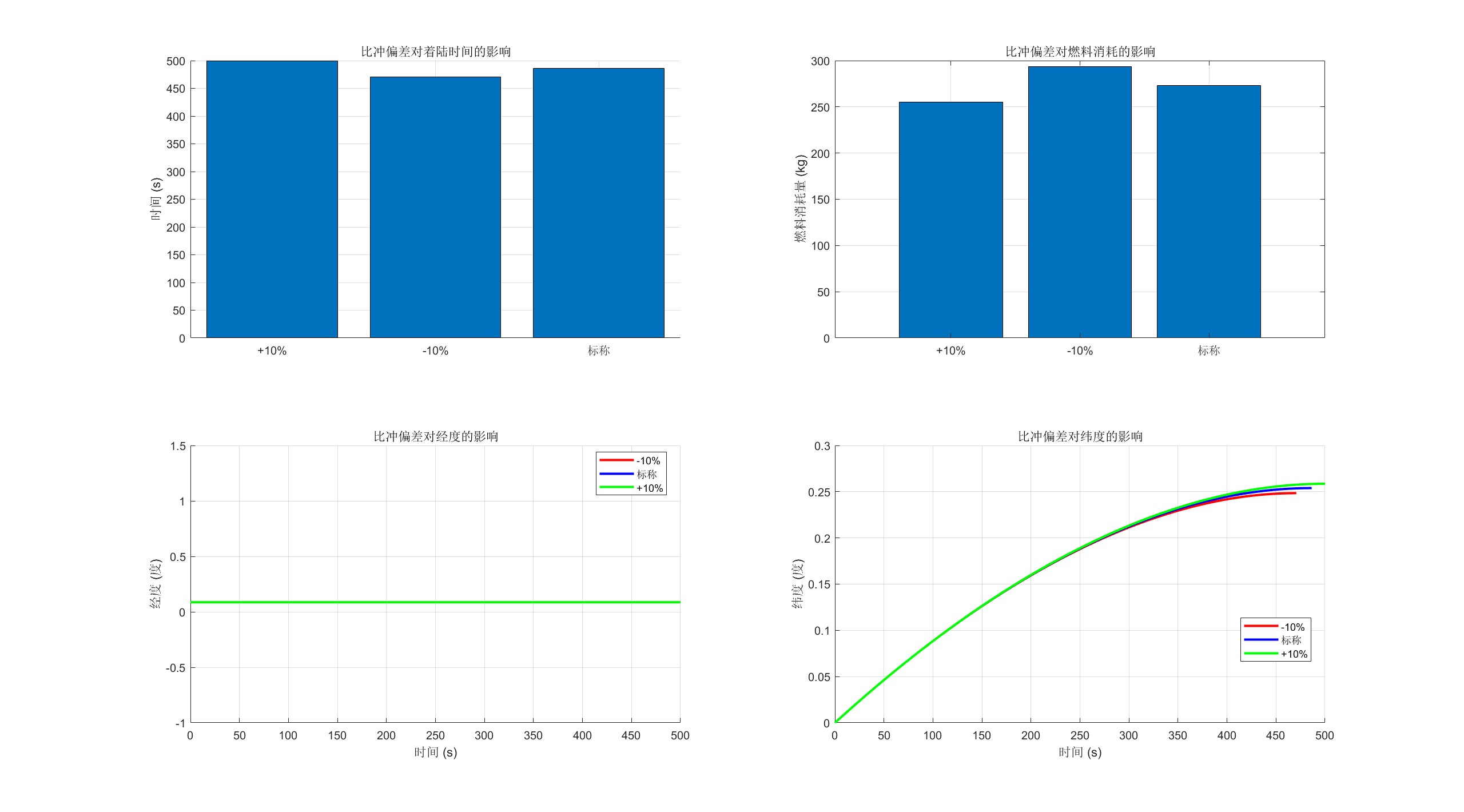
**B. 完整仿真结果图集**



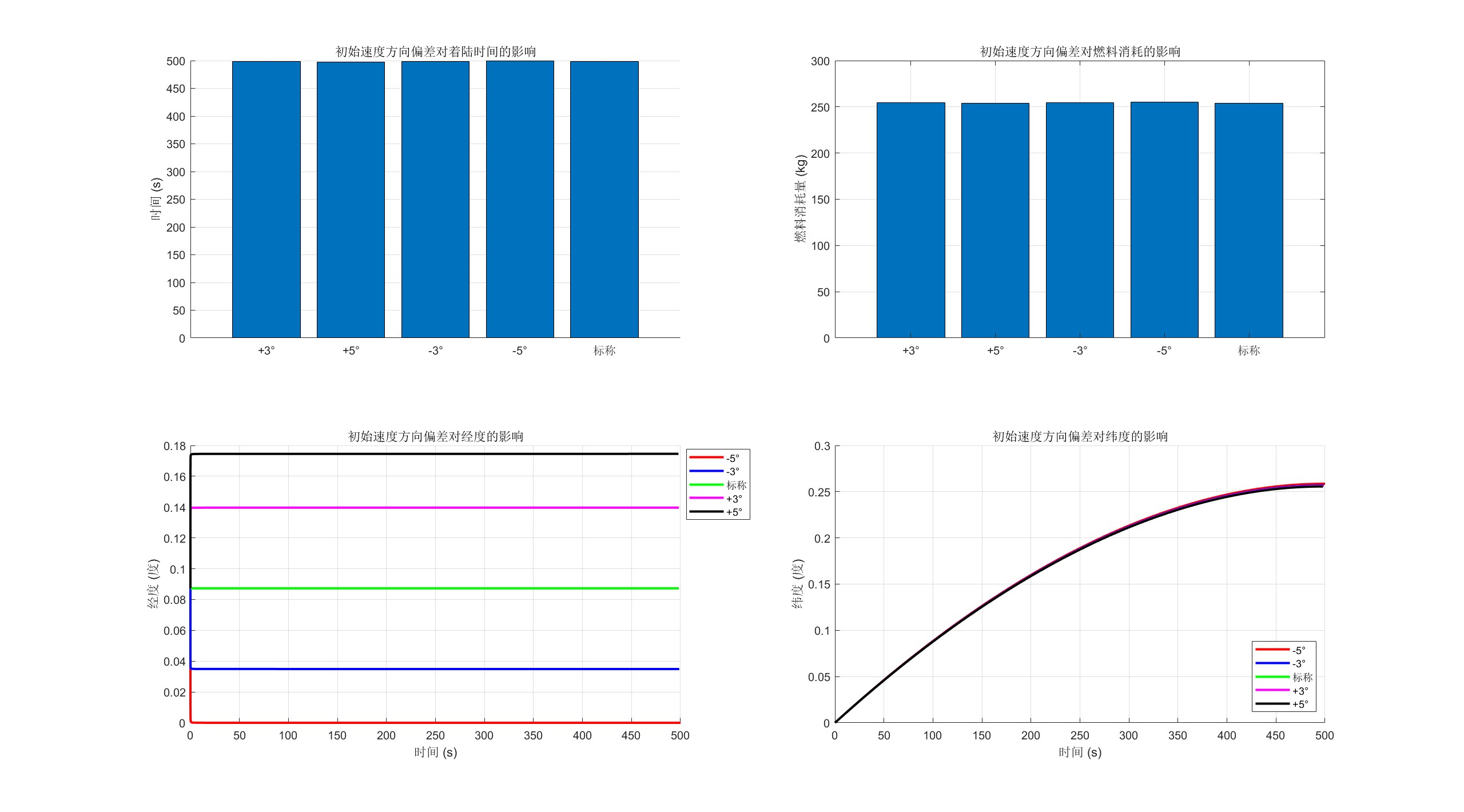
图B-1 标称工况着陆轨迹



图B-2 推力偏差影响分析



图B-3 比冲偏差影响分析



图B-4 初始速度偏差偏差影响分析