# Final Exam
# AE 6511 Optimal Guidance and Control

## Instructions

1. This is an open-book, open-notes exam
2. You will need to upload the exam to Canvas by 5:00pm ET on December 9, 2021
3. No collaboration of any kind between students is allowed
4. Include all intermediate steps for full credit. Box your answer and state the solution clearly
5. Points will be subtracted for sloppiness
6. Total number of points is 100

Good luck!

**Student Agreement**

I certify that I have read and understand the above ground rules for the exam. I also understand that any violations of these rules or those of the Georgia Tech Honor Code will be treated as a violation of the Honor Code.

Name and Signature: Tomoki Koike

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | |

December 9, 2021

## Table of Contents

## Problem 1

(10pts) Consider the problem of maximizing the range of a rocket plane in horizontal flight. The equations of motion of the rocket are

$$\begin{aligned}\frac{dx}{dt} &= v \\ \frac{dv}{dt} &= \frac{1}{m}\left(c\beta - D\right) \\ \frac{dm}{dt} &= -\beta\end{aligned}$$

where $c$ is a positive constant, $D$ is a drag which depends on speed $v$ and lift $L$ as follows

$$D = Av^2 + BL^2, \qquad A \text{ and } B = \text{const.} > 0$$

and the lift $L$ is adjusted to balance the weight $mg$. The thrust depends on the rate of fuel consumption by

$$T = -c\frac{dm}{dt}.$$

We wish to determine how the thrust $T$ must be varied in order to maximize the range $x(t_1) - x(0)$. Deduce the extremals for this problem. (Hint: Use mass as the independent variable and find the optimal velocity profile as the function of the mass).

---

**Solution:**
The cost function for this problem is

$$\min J = -\int_0^{t_1} \dot{x}dt = -\int_0^{t_1} vdt.$$

From what we are given we can rewrite the equation to get an expression for the velocity as follows

$$\begin{aligned}m\dot{v} &= -c\dot{m} - Av^2 - Bm^2g^2 \\ Av^2 &= -c\dot{m} - Bm^2g^2 - m\dot{v} \\ v &= \sqrt{\frac{-c\dot{m} - Bm^2g^2 - m\dot{v}}{A}} > 0.\end{aligned}$$

Now we use the calculus of variation to solve the optimization problem for

$$F(v, \dot{v}, m, \dot{m}, t) = -v = -\sqrt{\frac{-c\dot{m} - Bm^2g^2 - m\dot{v}}{A}}.$$

If we take the derivative of this function with respect to $\dot{v}$ and $\dot{m}$ we have

$$F_{\dot{v}\dot{v}} = \frac{m^2}{4A^2}\left(\frac{-c\dot{m} - Bm^2g^2 - m\dot{v}}{A}\right)^{-\frac{3}{2}}$$
$$F_{\dot{m}\dot{m}} = \frac{c^2}{4A^2}\left(\frac{-c\dot{m} - Bm^2g^2 - m\dot{v}}{A}\right)^{-\frac{3}{2}}$$

which implies that $F_{\dot{v}\dot{v}}, F_{\dot{m}\dot{m}} > 0$, and therefore, the Legendre condition is satisfied. Next, we compute the Euler Lagrange equations for this multivariate function as follows.

$$\frac{d}{dt}\left(\frac{\partial F}{\partial \dot{v}}\right) - \frac{\partial F}{\partial v} = 0$$
$$\frac{d}{dt}\left(\frac{\partial F}{\partial \dot{m}}\right) - \frac{\partial F}{\partial m} = 0,$$

which results in

$$m^2\ddot{v} + cm\ddot{m} = 2c\dot{m} + m\dot{v}\dot{m} \quad \cdots \quad (1)$$
$$c^2\ddot{m} + cm\ddot{v} + 2m\dot{v}^2 + 6Bcg^2m\dot{m} + 6Bg^2m^2\dot{v} + 3c\dot{v}\dot{m} + 4B^2g^4m^3 = 0 \quad \cdots \quad (2)$$

If we calculate $((2) \times m - (1) \times c)$ we get

$$2m^2\dot{v}^2 + 6Bcg^2m^2\dot{m} + 6Bg^2m^2\dot{v} + 3cm\dot{v}\dot{m} + 4B^2g^4m^4 + 2c^2\dot{m}^2 + cm\dot{v}\dot{m} = 0$$
$$2\left(m^2\dot{v}^2 + c^2\dot{m}^2 + 2cm\dot{v}\dot{m}\right) + 6Bg^2m^2\left(m\dot{v} + c\dot{m}\right) + 4B^2g^4m^4 = 0$$
$$\left(m\dot{v} + c\dot{m}\right)^2 + 3Bg^2m^2\left(m\dot{v} + c\dot{m}\right) + 2B^2g^4m^4 = 0$$
$$\left[\left(m\dot{v} + c\dot{m}\right) + Bg^2m^2\right]\left[\left(m\dot{v} + c\dot{m}\right) + 2Bg^2m^2\right] = 0$$

Thus, the answer could be either

$$m\dot{v} + c\dot{m} + Bg^2m^2 = 0$$
$$m\dot{v} + c\dot{m} + 2Bg^2m^2 = 0$$

However, since if we plug $m\dot{v} = -c\dot{m} - Av^2 - Bm^2g^2$ into the first equation we have

$$Av^2 = 0$$

which is not appropriate since $v \neq 0$, and therefore the second one is the only option. The second equation gives us

$$\left(-c\dot{m} - Av^2 - Bm^2g^2\right) + c\dot{m} + 2Bm^2g^2 = 0$$
$$Av^2 = Bm^2g^2$$

$$\boxed{v = mg\sqrt{\frac{B}{A}}.}$$

If we take the derivative of this equation of $v(m)$ we get

$$\dot{v} = \dot{m}g\sqrt{\frac{B}{A}}$$

$$c\beta - D = -\beta g\sqrt{\frac{B}{A}}$$

$$c\beta - A\left(mg\sqrt{\frac{B}{A}}\right)^2 - Bm^2g^2 = -\beta g\sqrt{\frac{B}{A}}$$

$$\dot{m}\left(c + g\sqrt{\frac{B}{A}}\right) = -2Bm^2g^2$$

$$\frac{dm}{m^2} = -\frac{2Bg^2}{c + g\sqrt{\frac{B}{A}}}dt$$

$$-\frac{1}{m} = -\frac{2Bg^2}{c + g\sqrt{\frac{B}{A}}}t + c$$

$$m = \frac{1}{\frac{2Bg^2}{c + g\sqrt{\frac{B}{A}}}t - c}.$$

Let, $m(0) = m_0$ and then we have,

$$\boxed{m = \frac{m_0}{\frac{2Bg^2m_0}{c + g\sqrt{\frac{B}{A}}}t + 1}.}$$

Now since

$$\dot{m} = -\frac{2Bg^2}{c + g\sqrt{\frac{B}{A}}}m^2 \qquad \text{and} \qquad T = -c\dot{m},$$

we can compute the thrust to be

$$\boxed{T(t) = \frac{2Bg^2c\left(c + g\sqrt{\frac{B}{A}}\right)m_0^2}{\left(2Bg^2m_0t + c + g\sqrt{\frac{B}{A}}\right)^2}.}$$

Finally, to compute $x$, we solve

$$\dot{x} = v = mg\sqrt{\frac{B}{A}}$$

$$\therefore x = \frac{g\sqrt{\frac{B}{A}}\left(c + g\sqrt{\frac{B}{A}}\right)}{2Bg^2m_0} \ln\left(2Bg^2m_0t + c + g\sqrt{\frac{B}{A}}\right).$$

Thus, the extremal of $x(t_1) - x(0)$ becomes

$$\boxed{\max(x(t_1) - x(0)) = \frac{g\sqrt{\frac{B}{A}}\left(c + g\sqrt{\frac{B}{A}}\right)}{2Bg^2m_0}\left(\ln\left(2Bg^2m_0t_1 + c + g\sqrt{\frac{B}{A}}\right) - \ln\left(2Bg^2m_0\right)\right).}$$

## Problem 2

(10pts) Consider the linear dynamic system that is self-adjoint, that is,

$$\dot{x} = Fx + u, \qquad F = -F^T.$$

Determine $u(t)$ subject to the constraint $u(t) \in U = \left\{u : \|u\|^2 = 1\right\}$ such that $x(t_f) = 0$, where $t_f$ is a minimum. Find the feedback solution; i.e., express $u(t)$ explicitly as a function of $x$ and $t$.

---

**Solution:**
The cost for this problem becomes

$$\min J = \int_0^{t_f} 1dt$$

and thus, the Hamiltonian for this problem is

$$H = 1 + \lambda^T (Fx + u) + \mu(u^T u - 1).$$

The costate for this problem is

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = -(\lambda^T F)^T = F\lambda$$

and the optimal control becomes

$$\frac{\partial H}{\partial u} = \lambda^T + 2\mu u^T = 0$$
$$u = -\frac{1}{2\mu}\lambda.$$

If the matrix $F$ is self-adjoint, this implies that the matrix exponential $e^F$ is orthogonal. Therefore, if we solve for $\lambda$ using the costate equation we can say

$$\lambda = e^{F(t-t_0}\lambda(0) = \Phi\lambda_0$$

where $\Phi$ denotes the state transition matrix. Now we can express the optimal control as

$$u = -\frac{1}{2\mu}\Phi(t)\lambda_0$$

which is then substituted into the dynamics

$$\dot{x} = Fx - \frac{1}{2\mu}\lambda_0 = \Phi(t)x_0 + \int_0^{t_f} e^{F(t_f-\tau)}d\tau.$$

Using the trick of $\Phi(t,0) = \Phi(t)$ and $\Phi(0,\tau) = \Phi^{-1}(\tau,0)$ where $u(\tau) = -\Phi(\tau)\lambda_0/2\mu$ we can rewrite the above as

$$x(t) = \Phi(t)x_0 + \Phi(t)\int_0^t \Phi^{-1}(\tau)u(\tau)d\tau$$
$$x(\lambda) = \Phi(t)x_0 - \Phi(t)\int_0^t \Phi^{-1}(\tau)\Phi(\tau)\frac{1}{2\mu}\lambda_0 d\tau$$
$$x(\lambda) = \Phi(t)\left[x_0 - \frac{\lambda_0}{||\lambda_0||}t\right]$$

This is because

$$||u||^2 = ||-\frac{1}{2\mu}\lambda||^2 = |\frac{1}{2\mu}|^2||\lambda_0||^2\Phi^T\Phi$$
$$\because ||u||^2 = 1, \qquad \Phi^T\Phi = 1$$
$$\frac{1}{2\mu} = \frac{1}{||\lambda_0||}.$$

Since at $t = t_f$ $x(t_f) = 0$ and we know that $\Phi(t)$ is invertible. This allows us to deduce

$$x_0 = t_f\frac{\lambda_0}{||\lambda_0||}$$
$$||x_0|| = t_f$$

Then we know that

$$x(t) = \Phi(t)\left[x_0 - \frac{x_0}{t_f}t\right]$$
$$= \Phi(t)\frac{x_0}{||x_0||}(||x_0|| - t)$$

Which then becomes

$$||x(t)|| = ||\Phi(t)(||x_0|| - t)||$$
$$= (||x_0|| - t)||\Phi(t)||$$
$$\because ||\Phi(t)|| = 1 \qquad \text{due to orthogonality}$$
$$= ||x_0|| - t$$

Hence,

$$\frac{x(t)}{||x(t)||} = \Phi(t)\frac{x_0}{||x_0||} = \Phi(t)\frac{\lambda_0}{||\lambda_0||} = -u.$$

The answer is

$$\boxed{u = -\frac{x(t)}{||x(t)||}.}$$

The paper by Dr. Michael Athans et al. (1964) was made use of for this problem [1].

## Problem 3

(10pts) Consider the minimum fuel problem with cost

$$J(u) = \int_0^{t_f} |u(t)| dt$$

with the scalar dynamics $\dot{x(t)} = -ax(t) + u(t)$, the control constraint

$$|u(t)| \le u_{max}$$

and boundary conditions $x(0) = x_0$ and $x(t_f) = 0$. Here, $a > 0$ and $x_0$, $t_f$ fixed.

(a) Determine the set of initial conditions $x_0$ for which the endpoint constraint $x(t_f) = 0$ can be satisfied. Your answer should be in the form of an inequality.

(b) Use the minimum principle to determine the optimal control, including an explicit expression for the switch times. (Hint: The optimal control is a coast−burn strategy.)

(c) Show that the optimal control can be written as a time-varying switching feedback control law $u(t) = \phi(x(t), t)$. Sketch some optimal trajectories to demonstrate the control law.

---

**Solution:**
(a) The Hamiltonian is

$$H = |u(t)| + \lambda(-ax + u).$$

The costates are

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = a\lambda,$$

which gives us

$$\lambda^* = Ce^{at}.$$

For minimum $x_0$ we can expect $u(t) = u_{max} =$ const. which implies

$$\dot{x} = -ax + u_{max} \qquad \text{and } x(t_f) = 0.$$

For maximum $x_0$ we can expect $u(t) = u_{min} =$ const. which implies

$$\dot{x} = -ax - u_{max} \qquad \text{and } x(t_f) = 0.$$

Then solving these two differential equations we obtain the following inequality

$$\boxed{(1 - e^{at_f})\frac{u_{max}}{a} \le x_0 \le (e^{at_f} - 1)\frac{u_{max}}{a}.}$$

(b) Plugging in the results from problem (a) into the Hamilonian we have

$$H = \begin{cases} -\lambda_0 e^{at} a x + (1 + \lambda_o e^{at}) u(t) & (u > 0) \\ -\lambda_0 e^{at} a x + (-1 + \lambda_o e^{at}) u(t) & (u < 0) \end{cases}$$

When the coasting strategy is implemented and we have

$$u^*(t) = 0,$$

otherwise the burn strategy is applied, in which the optimal control becomes

$$\begin{aligned} u^*(t) &= -\operatorname{sgn}(-1 + \lambda_0 e^{at}) u_{max} \\ &= -\operatorname{sgn}(-1 + \lambda^*) u_{max}. \end{aligned}$$

Hence, the optimal control is expressed as

$$\boxed{u^* = \begin{cases} -u_{max} & \text{if } \lambda^* > 1 \\ 0 & \text{if } |\lambda^*| \leq 1 \\ u_{max} & \text{if } \lambda^* < -1 \end{cases}}$$

The switching times $t_i \in \mathbb{R}$ are times that satisfy

$$\boxed{|\lambda^*(t_i)| = |\lambda_0 e^{at_i}| = 1.}$$

(c) From the switching time we have

$$\begin{aligned} |\lambda_0 e^{at}| &= 1 \\ \lambda_0 e^{at} &= \pm 1 \\ e^{at} &= \pm \frac{1}{\lambda_0} \\ \therefore t &= \frac{1}{a} \ln \left( \pm \frac{1}{\lambda_0} \right) \end{aligned}$$

This is

$$t_s = \frac{1}{a} \ln \left( \frac{1}{|\lambda_0|} \right).$$

Hence, if we combine all the analysis from Problem (a)-(c) we have the following control

$$\boxed{u(t) = \begin{cases} (x_0 > 0) \left\langle \begin{array}{ll} 0 & t \in [0, t_s] \\ u_{max} & t \in [t_s, t_f] \end{array} \right. \\ \\ (x_0 < 0) \left\langle \begin{array}{ll} 0 & t \in [0, t_s] \\ -u_{max} & t \in [t_s, t_f] \end{array} \right. \end{cases}}$$

Example sketches are as follows.

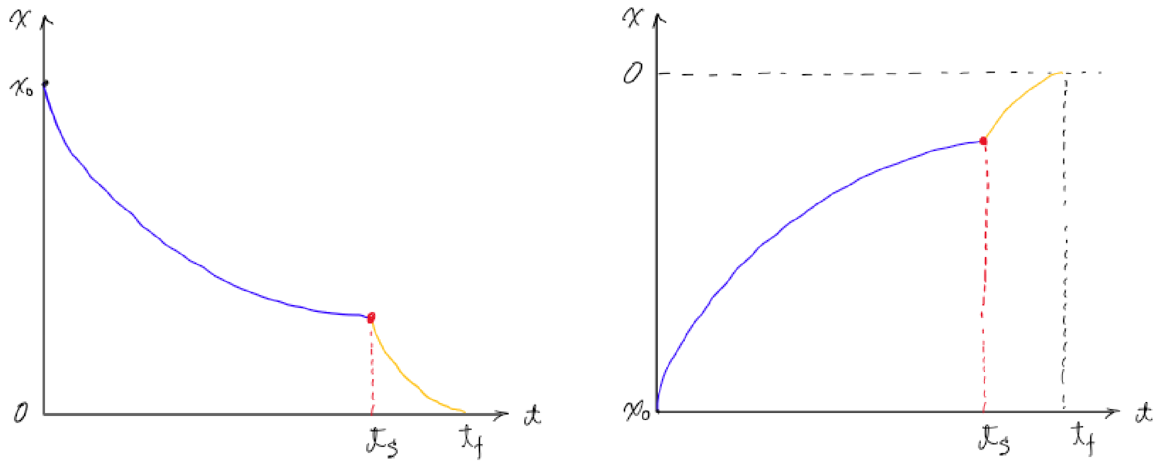


Figure 1: Example sketches of coast-burn control

## Problem 4

(10pts) Consider the problem

$$\min J(u) = \frac{1}{2}\int_0^T x_1^2 dt$$

subject to the constraints

$$\dot{x}_1 = x_2 + u$$
$$\dot{x}_2 = -u$$

where

$$|u| \leq 1$$

and with boundary conditions

$$x_1(0) = 0, \qquad x_2(0) = -0.5, \qquad x_1(T) = x_2(T) = 0$$

and the final time $T$ is free.

(a) What is the optimal trajectory?

(b) Can we reach the origin using only bang-bang control?

(c) Compare the cost of the optimal strategy versus the pure bang-bang strategy.

(d) What is the terminal time for the two control strategies?

---

**Solution:**
(a) The Hamiltonian for this problem is

$$H = \frac{1}{2}x_1^2 + \lambda_1(x_2 + u) + \lambda_2(-u) = \frac{1}{2}x_1^2 + \lambda_1 x_2 + (\lambda_1 - \lambda_2)u.$$

The costate equations are

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x_1} = -x_1$$
$$\dot{\lambda}_2 = -\frac{\partial H}{\partial x_2} = -\lambda_1$$

and the optimal control becomes

$$\frac{\partial H}{\partial u} = \lambda_1 - \lambda_2 = 0.$$

Since $|u| \leq 1$ and the Hamiltonian is linear we can use Pontryagin's maximum principle to deduce the optimal control to be

$$u = -\operatorname{sgn}(\lambda_1 - \lambda_2).$$

When $u = -1$

$$\begin{aligned}\dot{x}_1 &= x_2 - 1\\ \dot{x}_2 &= 1\end{aligned}$$

solving these equations with the given initial conditions we get the trajectory

$$x_1 = 0.5(x_2 + 0.5)^2 - 1.5(x_2 + 0.5) \quad \cdots \quad (1)$$

Similarly, for $u = 1$ we have the trajectory

$$x_1 = -0.5(-x_2 - 0.5)^2 + 0.5(-x_2 - 0.5) \quad \cdots \quad (2)$$

Now, if $\lambda_1 - \lambda_2 = 0$ is true, the optimal control will involve singularity. For this case, the optimal control will be different from the above.

$$\begin{aligned}\frac{dH_u}{dt} &= \dot{\lambda_1} - \dot{\lambda_2} = -x_1 + \lambda_1 \quad \rightarrow \quad \therefore \lambda_1 = x_1\\ \frac{d^2H_u}{dt^2} &= -\dot{x}_1 + \dot{\lambda_1} = -x_2 - u - x_1 \quad \rightarrow \quad \therefore u_s = -x_1 - x_2.\end{aligned}$$

Thus, the singular control becomes

$$u_s = -(x_1 + x_2),$$

where the singular arc is

$$\frac{1}{2}x_1^2 + \lambda_1 x_2 = \frac{1}{2}x_1^2 + x_1 x_2 = \text{const.}$$

To verify that the singular control is optimal we check the generalized Legendre-Clebsh condition

$$(-1)\frac{\partial}{\partial u}\left[\frac{d^2}{dt^2}\left(\frac{\partial H}{\partial u}\right)\right] = -1(-1) = 1 > 0.$$

Which indicates that the singular control is optimal. From the initial conditions we know that the singular arc becomes

$$\begin{aligned}\frac{1}{2}x_1^2 + x_1 x_2 &= 0\\ x_1(\frac{x_1}{2} + x_2) &= 0\end{aligned}$$

$$\therefore \begin{cases} x_1 = 0 & \cdots \quad (3)\\ x_2 = -0.5x_1 & \cdots \quad (4)\end{cases}$$

Now if we plot the trajectories

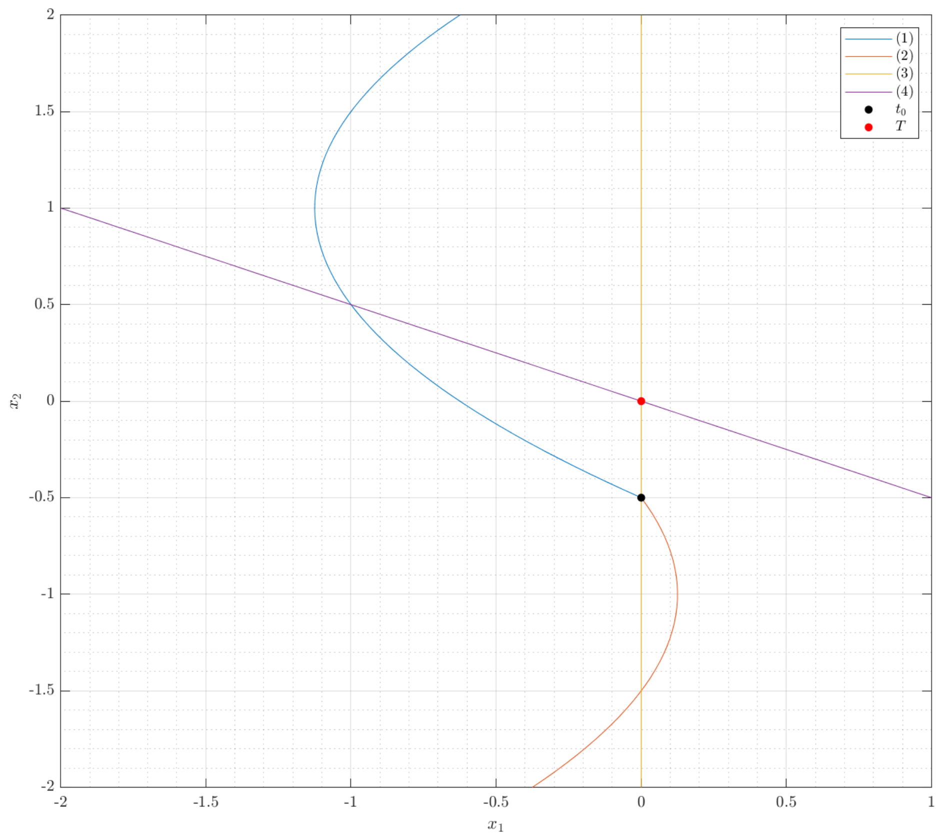


Figure 2: Bang-bang and singular trajectories of the problem

we can observe that the optimal trajectory is first along trajectory (1) and the second section along trajectory (4). If we calculate the intersection of these trajectories we obtain $(-1, 0.5)$ at $t = 1$. With the singular control we have

$$\dot{x}_1 = -x_1$$
$$\dot{x_2} = x_1 + x_2$$

with boundary conditions of $x_1(1) = -1$ and $x_2(1) = 0.5$ which becomes

$$x_{1s} = -2.7183e^{-t}$$
$$x_{2s} = 1.3591e^{-t}$$

and therefore, the final time $T$ is $\infty$ for it to reach zero. Hence, the optimal trajectory is

$$\boxed{\begin{aligned} x_1 &= \frac{1}{2}(x_2 - 1)^2 - 1.1250 \qquad && t \in [0, 1] \\ x_1 &= -2x_2 && t \in [1, \infty] \end{aligned}}$$

(b) To find a bang-bang arc that arrives at the origin we can say that the trajectory initially follows the trajectory (1) computed in Problem (a) and then follows a different arc for $u = 1$ that goes to the origin. To find this second trajectory with $u = 1$ we first define it as

$$\begin{aligned} x_1 &= -0.5t^2 + (c_1 + 1)t + c_2 \\ x_2 &= -t + c_1 \end{aligned}$$

and

$$x_1 = -0.5(-x_2 + c_1)^2 + (c_1 + 1)(-x_2 + c_1) + c_2.$$

If this trajectory goes through the origin and at some time $t = \tau$ intersects with the trajectory (1) we can find the constants $c_1$ and $c_2$. At this certain time $\tau$ we equate the time parameterized equations for the one above and trajectory (1).

$$\begin{gathered} -\tau + c_1 = \tau - 0.5 \\ -0.5\tau^2 + (c_1 + 1)\tau + c_2 = 0.5\tau^2 - 1.5\tau \end{gathered}$$

and we know that this new trajectory goes through the origin which gives

$$0 = -0.5 + 0.5(c_1 + 1)^2 + c_2.$$

Solving these we find the following,

$$c_1 = 2.0811, \quad c_2 = -4.2467, \quad \tau = 1.2906,$$

and the trajectories are characterized as

$$\begin{aligned} x_1(t) &= -0.5t^2 + 3.0811t - 4.2467 \\ x_2(t) &= -t + 2.0811 \end{aligned}$$

which becomes

$$x_1 = -0.5(-x_2 - 1)^2 + 0.4999 \quad \cdots \quad (5)$$

Adding this new trajectory we have the following plot.

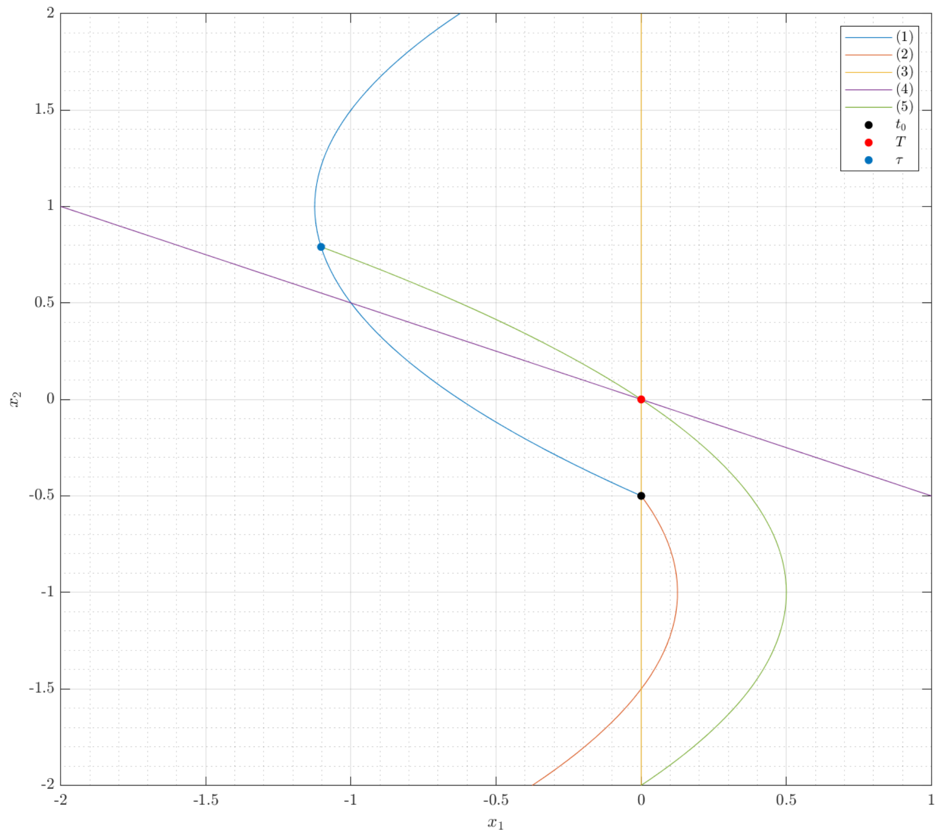


Figure 3: Updated Bang-bang and singular trajectories of the problem

Hence, we can conclude that even the pure bang-bang control does arrive at the origin.

(c) The second arc for the pure bang-bang scheme arrives at the origin at time $\tau_0$ which is

$$\tau_0 = 2.0812.$$

Then the cost for the optimal control becomes

$$\begin{aligned} J_{opt} &= 0.5 \int_0^1 \left(0.5t^2 - 1.5t\right)^2 dt + 0.5 \int_1^\infty \left(-2.7183e^{-t}\right)^2 dt \\ &= 0.4625. \end{aligned}$$

The cost for the pure bang-bang scheme becomes

$$\begin{aligned} J_{bb} &= 0.5 \int_0^\tau \left(0.5t^2 - 1.5t\right)^2 dt + 0.5 \int_\tau^{\tau_0} \left(-0.5t^2 + 3.0811t - 4.2467\right)^2 dt \\ &= 0.5144. \end{aligned}$$

Hence,

$$\boxed{\begin{gathered} J_{opt} = 0.4625 \\ J_{bb} = 0.5144 \\ \\ J_{bb} \text{ is } 11.2104\% \text{ larger than } J_{opt} \end{gathered}}$$

(d) The terminal time for each strategy has already been calculated in Problems (a)-(c). Comparing the two we know the following.

$$\boxed{\begin{gathered} t_{opt} = \infty \\ t_{bb} = \tau_0 = 2.0812 \end{gathered}}$$

The optimal strategy will not reach the optimal value in a finite amount of time which is not ideal in reality; whereas, the bang-bang strategy is able to reach the origin in a fairly short amount of time, 2.0812 seconds.

## Problem 5

(10pts) Find the path of minimum time connecting two points on the surface of the Earth through a tunnel in the Earth. The tunnel is assumed to be evacuated, and friction is negligible. The only force acting on the particle is gravity. Note that the gravitational force per unit mass inside the Earth is directed radially toward the center of the Earth and increases *linearly* with the radius from zero at the center. (Hint: use spherical coordinates)

---

**Solution:**
First we define the following constants

$M :=$ mass of the Earth , $\quad R :=$ radius of the Earth, $\quad \rho :=$ mass density of the Earth

The integrated mass at a distance of $r$ from the center of the Earth is to its surface is

$$M = \int_0^R 4\pi r^2 \rho dr = \frac{4}{3}\pi R^3 \rho,$$

and we have

$$\rho = \frac{3M}{4\pi R^3}.$$

Expressing the acceleration due to gravity at radius $r$ we have

$$F(r) = \frac{GMm}{r^2} = \frac{GM}{R^3} r.$$

If the mass of the particle is $m$ we can integrate the force to obtain the potential energy

$$U(r) = \int_0^r F(r)dr = \frac{GMm}{2R^3} r.$$

To find the velocity expression we will use the conservation of energy of the particle

$$E = T + U = \frac{1}{2}mv^2 + \frac{GMm}{2R^3} r^2,$$

now if $v = 0$ at $r = R$ we can solve the above equation to find $v$ to be

$$v = \sqrt{\frac{g(R^2 - r^2)}{R}}$$

where the gravitational acceleration $g$ denotes

$$g = \frac{GM}{R^2}.$$

To solve the minimization problem for the particle to slide through a tunnel connecting two locations on the surface of the Earth - $A$ and $B$ - we first define the time $t$ to be

$$T = \int_A^B \frac{1}{v} ds,$$

where $ds$ is the distance along the path. To transform what we have so far into the polar coordinates we use

$$x = r\cos\theta$$
$$y = r\sin\theta$$
$$r^2 = x^2 + y^2$$

The distance of the tunnel becomes

$$ds = \sqrt{\left(\frac{dx}{d\theta}\right)^2 + \left(\frac{dy}{d\theta}\right)^2} = \sqrt{r^2 + r'^2}$$

where $r' = dr/d\theta$. The if we say the angle between the locations $A$ and $B$ is $\theta_{AB}$ we can rewrite $T$ as

$$T = \int_0^{\theta_{AB}} \sqrt{\frac{(r^2 + r'^2)R}{g(R^2 - r^2)}}$$

which is subject to the constraints of $r(0) = R$ and $r(\theta_{AB}) = R$. Since this integral is independent of the variable $\theta$ we are able to use the special form (Beltrami identity) for the Euler-Lagrange equation

$$f - r'\frac{\partial f}{\partial r'} = C = \text{ const.}$$

which with some simplifications reduces the following expression

$$\left(\frac{r^2}{r^2 - r'^2}\right)\sqrt{\frac{R(r'^2 - r^2)}{g(r^2 - R^2)}} = C$$

now if we solve this for $r'$ we obtain

$$r' = \pm\left(\frac{r}{\sqrt{2}C}\right)\sqrt{2C^2 + \frac{r^2}{g(r-R)} - \frac{r^2}{g(r+R)}}.$$

This can be integrated and converted back to the Cartesian coordinates as follows

$$x(t) = (R-b)\cos(t) + b\cos\left(\frac{R-b}{b}t\right)$$
$$y(t) = (R-b)\sin(t) - b\sin\left(\frac{R-b}{b}t\right)$$

in which this can be identified as a hypocycloid where $b$ indicates the radius of an inner circle. Now if we convert the expression of $T$ back to the Cartesian we have

$$T = \int_A^B \sqrt{\frac{(x'^2 + y'^2)R}{g(R^2 - (x^2 + y^2))}}$$

and since this equation does not depend on time we can use the Beltrami identity for the Euler-Lagrange equation

$$f - x'\frac{\partial f}{\partial x'} = C_x = \left(\frac{y'^2}{y'^2 - x'^2}\right)\sqrt{\frac{R(x'^2 - y'^2)}{g(R^2 - x^2 - y^2)}}$$
$$f - y'\frac{\partial f}{\partial y'} = C_y = \left(\frac{x'^2}{x'^2 - y'^2}\right)\sqrt{\frac{R(x'^2 - y'^2)}{g(R^2 - x^2 - y^2)}}$$

Adding the two above equations and squaring them gives

$$R(x'^2 + y'^2) = gC^2(R^2 - x^2 - y^2)$$

where

$$C = C_x + C_y.$$

Now if we substitute the derivatives of $x(t)$ and $y(t)$ (parametric expression of the hypocycloid) into this equation, we obtain an expression of $C$ in terms of $b$

$$C = \sqrt{\frac{R(R-b)}{gb}}.$$

Now we have all the necessary ingredients to solve the minimization problem. Since

$$T = \int_A^B C dt = \int_B^A \sqrt{\frac{R(R-b)}{gb}} dt$$

and that the time derivative $dt$ can be found in terms of $b$

$$r^2 = x^2 + y^2 = 2b^2 - 2bR + R^2 + 2b(R-b)\cos\left(\frac{Rt}{b}\right).$$

A single tunnel connecting two locations on the surface of the Earth is only one turn or cycle of the hypocycloid, that is,

$$\cos\left(\frac{Rt}{b}\right) = 1$$
$$\to \frac{Rt}{b} = 2\pi$$
$$\therefore dt = \frac{2\pi b}{R}.$$

From this we finally get the expression of

$$T = \frac{2\pi b}{R}\sqrt{\frac{R(R-b)}{gb}}$$

but to eliminate the term $b$ we use the characteristic of the hypocycloid that the number of cusps for a curve will be

$$N(\theta) = \frac{a\theta}{2\pi b}$$

where $a$ is the outer circle radius and $b$ is the inner circle radius. So we can say $R = a$ and for a single cusp we have the relation of

$$1 = \frac{R\theta_{AB}}{2\pi b}$$

and therefore,

$$b = \frac{R\theta_{AB}}{2\pi b}$$

Furthermore,

$$\theta_{AB} = \frac{L}{R}$$
$$\therefore b = \frac{L}{2\pi}$$

where $L$ is the distance from two locations $A$ and $B$ on the surface of the Earth. Hence, we have

$$\boxed{\begin{aligned} &T = \sqrt{\frac{L(2\pi R - L)}{gR}} \\ \\ R &:= \text{radius of the Earth} \\ L &:= \text{distance between locations A and B} \\ &\quad\ \text{on the surface of the Earth} \\ g &:= \text{gravitational acceleration} \end{aligned}}$$

## Problem 6

(10pts) Consider the system

$$\dot{x}_1 = x_2(t)$$
$$\dot{x}_2 = u_1(t)$$
$$\dot{x}_3 = x_4(t)$$
$$\dot{x}_4 = u_2(t)$$

with initial conditions

$$x_1(0) = 1, \quad x_2(0) = 0, \quad x_3(0) = -1, \quad x_4(0) = 0$$

terminal conditions

$$x_1(t_f) = x_3(t_f), \qquad x_2(t_f) = x_4(t_f)$$

with the final time free, and performance index

$$J(u_1, u_2) = \frac{1}{2}(x_2(t_f) - 2)^2 + \frac{1}{2}\int_0^{t_f} \left(u_1^2(t) + u_2^2(t)\right) dt.$$

Find the optimal control for this problem. Plot the trajectories of the system.

---

**Solution:**
In this problem we have a terminal cost of

$$\Phi(x(t_f, t_f) = \frac{1}{2}(x_2(t_f) - 2)^2$$

and a terminal constraint of

$$\Psi_1(x(t_f), t_f) = x_1(t_f) - x_3(t_f) = 0, \quad \Psi_2(x(t_f), t_f) = x_2(t_f) - x_4(t_f) = 0,$$

and the Hamiltonian becomes

$$H = \frac{1}{2}u_1^2 + \frac{1}{2}u_2^2 + \lambda_1 x_2 + \lambda_2 u_1 + \lambda_3 x_4 + \lambda_4 u_2.$$

The costate equations are

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x_1} = 0$$
$$\dot{\lambda}_2 = -\frac{\partial H}{\partial x_2} = -\lambda_1$$
$$\dot{\lambda}_3 = -\frac{\partial H}{\partial x_3} = 0$$
$$\dot{\lambda}_4 = -\frac{\partial H}{\partial x_4} = -\lambda_3$$

which gives us

$$\lambda_1 = c_1$$
$$\lambda_2 = -c_1 t + c_2$$
$$\lambda_3 = c_3$$
$$\lambda_4 = -c_3 t + c_4$$

The optimal control for this problem becomes

$$\frac{\partial H}{\partial u} = \begin{bmatrix} u_1 + \lambda_2 \\ u_2 + \lambda_4 \end{bmatrix}^T = 0$$

The transversality condition for this problem is

$$\begin{bmatrix} H(t_f) + \Phi_t(x(t_f), t_f) \\ -\lambda(t_f) + \Phi_x^T(x(t_f), t_f) \end{bmatrix} = \begin{bmatrix} \Psi_t^T(x(t_f), t_f) \\ \Psi_x^T(x(t_f), t_f) \end{bmatrix} \zeta$$

Which is

$$H(t_f) = 0$$

$$\begin{bmatrix} -\lambda_1(t_f) \\ -\lambda_2(t_f) + x_2(t_f) - 2 \\ -\lambda_3(t_f) \\ -\lambda_4(t_f) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix}.$$

This is reduced to a more clear form of the transversality condition of

$$H(t_f) = 0$$
$$\lambda_1(t_f) + \lambda_3(t_f) = 0$$
$$\lambda_2(t_f) + \lambda_4(t_f) = x_2(t_f) - 2$$

From all of these we know that

$$x_1 = \frac{c_1}{6}t^3 - \frac{c_2}{2}t^2 + c_5 t + c_6$$
$$x_2 = \frac{c_1}{2}t^2 - c_2 t + c_5$$
$$x_3 = \frac{c_3}{6}t^3 - \frac{c_4}{2}t^2 + c_7 t + c_8$$
$$x_4 = \frac{c_3}{2}t^2 - c_4 t + c_7$$

Now, we have four equations from the boundary conditions, 3 equations from the transversality condition, and 2 equations from the terminal constraint which gives us nine equations in total. Since, we have nine unknowns this problem is solvable. To plot the results we use `bvp4c()` of MATLAB to find the optimal states, costates, and controls (refer to the code in Problem 6: MATLAB Code). The results are as follows.

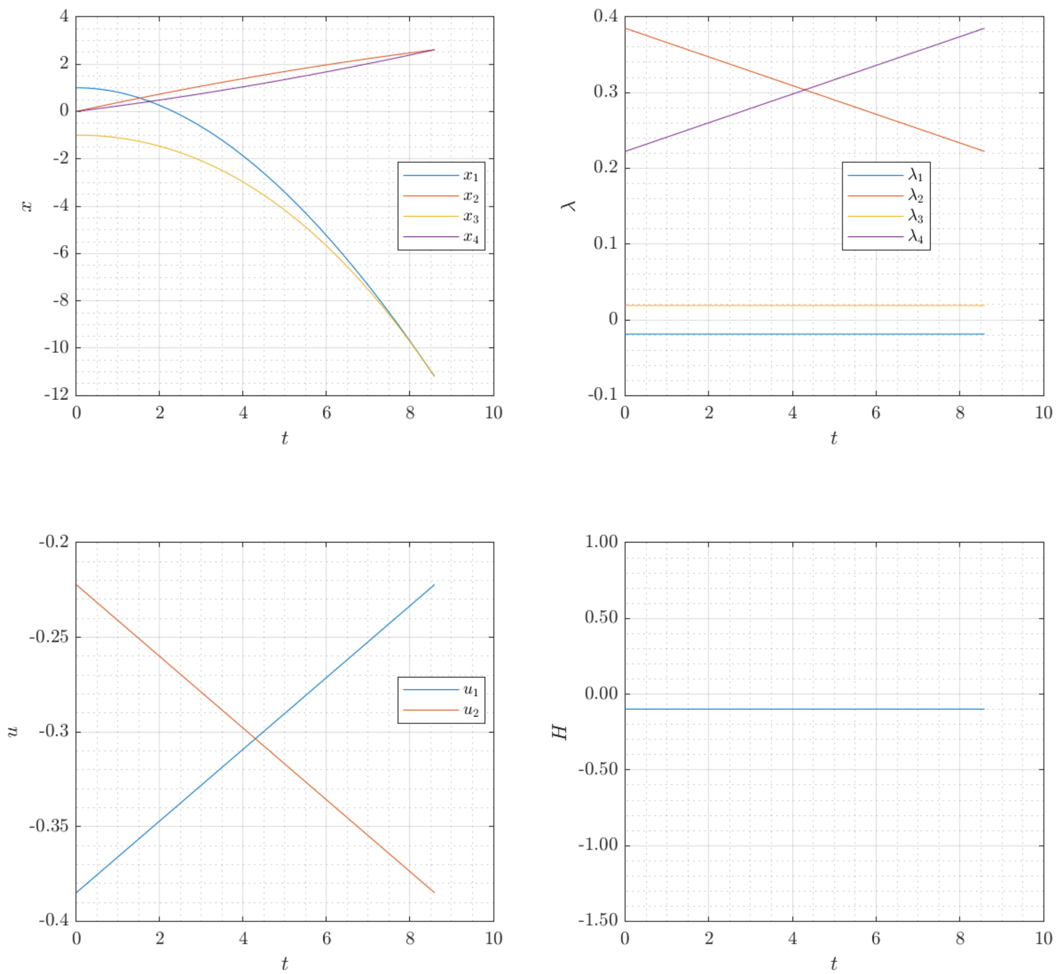


Figure 4: Numerical calculation of minimization problem: states (top left), costates (top right), controls (bottom left), and Hamiltonian (bottom right)

From Figure 4, we can see that the Hamiltonian is constant at a value approximate to zero. Finally, the minimized objective becomes

$$\boxed{\min J(u_1, u_2) = 0.9945.}$$

## Problem 7

(40pts) Consider the problem of the longitudinal flight of a hypersonic vehicle over a spherical Earth. The equations of motion are given by

$$\begin{aligned}
\dot{r} &= v \sin\gamma \\
\dot{\phi} &= \frac{v}{r}\cos\gamma \\
\dot{v} &= -\frac{D}{m} - \frac{\mu \sin\gamma}{r^2} \\
\dot{\gamma} &= \frac{L}{mv} - \frac{\mu\cos\gamma}{r^2 v} + \frac{v\cos\gamma}{r}
\end{aligned}$$

where $r$ is the distance from center of the Earth, $\phi$ is the longitude angle, $v$ is the velocity, and $\gamma$ is the flight-path angle. The control is the angle of attack $\alpha$. The lift and drag forces are given by

$$L = \frac{1}{2}\rho(r)v^2 S C_L$$

where $C_L = a_0 + a_1\alpha$, and

$$D = \frac{1}{2}\rho(r)v^2 S C_D$$

where $C_D = b_0 + b_1\alpha + b_2\alpha^2$ where $\alpha$ is in deg, and where we assume an exponential model of the density

$$\rho(r) = \rho_0 \exp\left(-\frac{r - r_0}{H_s}\right).$$

We wish to maximize the final value of the longitude, that is,

$$\max J = \phi(t_f)$$

where the final time $t_f$ is unspecified. The boundary conditions for this problem are given below

| State | Initial Value | Final Value | Units |
|---|---|---|---|
| $h$ | 121.9 | 30.48 | km |
| $\phi$ | -25 | - | deg |
| $v$ | 7,626 | 908.15 | m/s |
| $\gamma$ | -1.25 | [-6, 0] | deg |

The vehicle is subject to path constraint as follows:

$$\Lambda = k_\lambda \sqrt{\rho} v^3 \leq \Lambda_{max} \qquad \text{(Heating)}$$
$$q = \frac{1}{2}\rho v^2 \leq q_{max} \qquad \text{(Dynamic Pressure)}$$
$$n = \frac{\sqrt{L^2 + D^2}}{m} \leq n_{max} \qquad \text{(Normal Load)}.$$

The vehicle parameters and the aerodynamic and atmospheric model parameters are given in the tables below.

(a) Using a code of your choice, compute the optimal control and the corresponding optimal trajectory. For better numerical accuracy, you may want to non-dimensionalize your equations using as length $r_0$ the final altitude, and the time scale as $\tau = \sqrt{r_0/g}$.

| Parameter | Value | Units |
|---|---|---|
| $S$ | 149.3881 | $m^2$ |
| $m$ | 38,000 | kg |
| $\Lambda_{max}$ | $4\times10^5$ | $W/m^2$ |
| $q_{max}$ | 14,500 | $kg/ms^2$ |
| $n_{max}$ | $5g_0$ | $m/s^2$ |
| $k_\lambda$ | $9.4369\times10^{-5}$ | $kg^{0.5}/m^{1.5}$ |
| $a_0$ | -0.20704 | - |
| $a_1$ | 0.029244 | - |
| $b_0$ | 0.07854 | - |
| $b_1$ | $-0.61592\times10^{-2}$ | - |
| $b_2$ | $0.621408\times10^{-3}$ | - |
| $\mu$ | $3.986\times10^{14}$ | $m^3/s^2$ |
| $\rho_0$ | 1.225 | $kg/m^3$ |
| $H_s$ | 7,254.24 | m |
| $r_0$ | 6,371 | km |

(b) Plot the optimal control $\alpha(t)$ and the optimal trajectory $x(t) = (r(t), \phi(t), v(t), \gamma(t))$ for $0 \le t \le t_f$.

(c) Plot the time history of the Hamiltonian along with history of the co-states.

(d) Develop a neighboring guidance scheme to augment the nominal optimal control you computed from part (a).

(e) Demonstrate the benefits of the neighboring optimal guidance scheme by applying a small constant vertical wind disturbance acting on the vehicle at $t = 1000$sec. Plot the open-loop and closed-loop trajectories of the vehicle subject to perturbations for different values of the magnitude of the wind strength.

(f) Evaluate your neighboring guidance scheme on parametric variations of the nominal values of the aerodynamic parameters $a_0, a_1, b_0, b_1, b_2$ by introducing a perturbations to their nominal values. Which of these parameters has the largest effect on the resulting trajectories? Plot several trajectories (∼10-20) with ±10% random variations in these parameters from their nominal values above, and compare the resulting trajectories with and without the neighboring guidance scheme.

---

**Solution:**

(a) To solve this problem numerically MATLAB's GPOP software was used. The code is in Problem 7.1: MATLAB Code, so please refer to that.

(b) The optimal control and states are as follows.

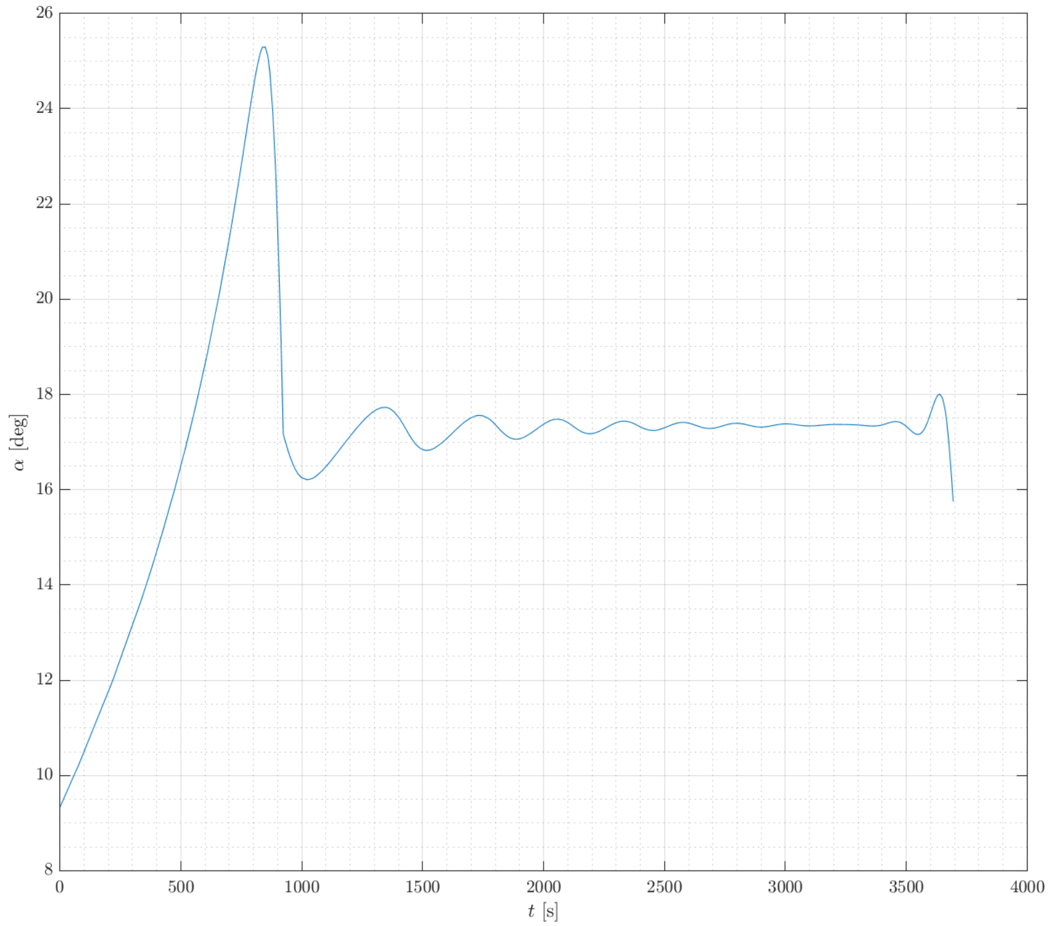


Figure 5: Optimal control

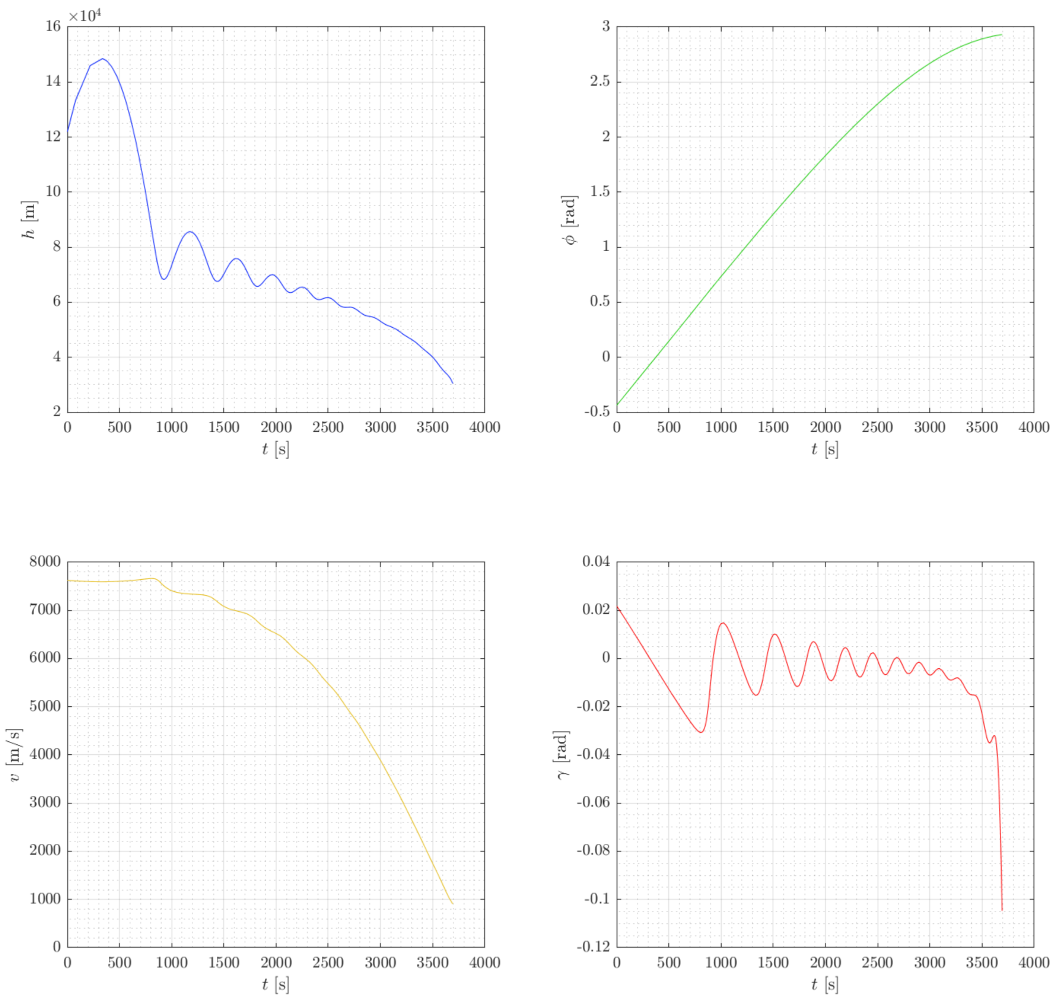


Figure 6: Optimal states

(c) Next, the time history of the Hamiltonian and the costates are as follows.

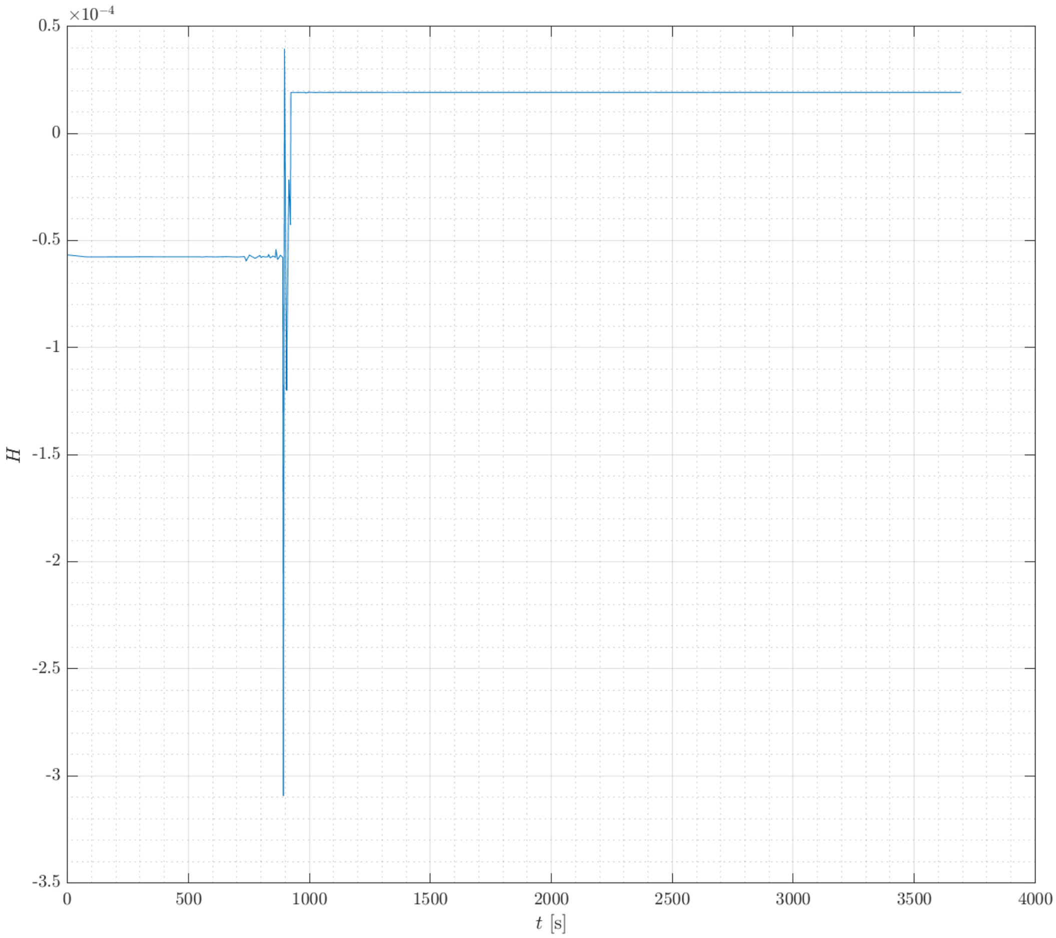


Figure 7: Optimized Hamiltonian

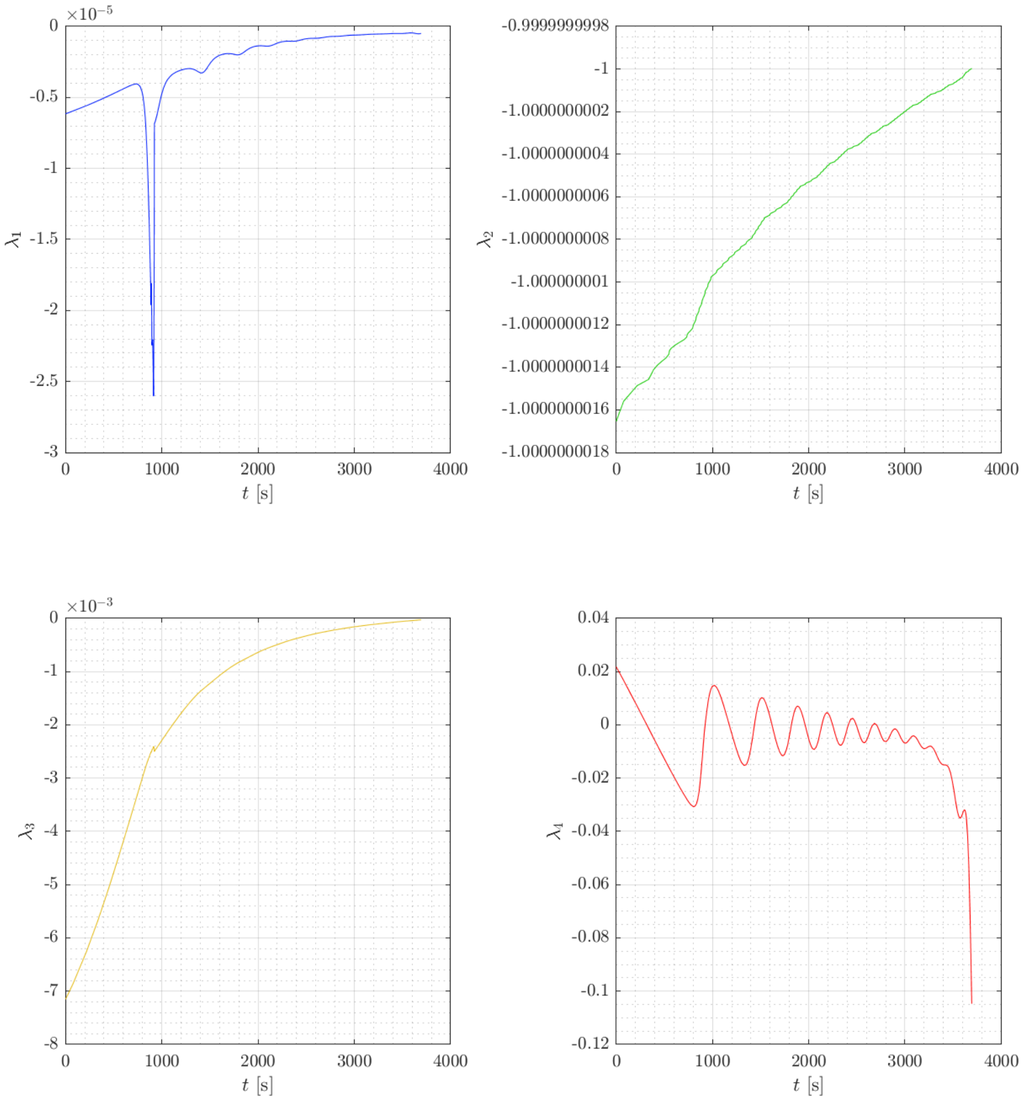


Figure 8: Optimal costates

(d) Considering small variations $\delta x, \delta u, \delta\lambda, dv$ in the problem we can define an accessory minimization problem as follows.

$$\delta^2 J_a = \frac{1}{2}\delta x^T(t_f)\phi_{xx}(t_f)\delta x(t_f) + \frac{1}{2}\int_0^{t_f} \begin{bmatrix}\delta x^T & \delta u^T\end{bmatrix} \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu}\end{bmatrix} \begin{bmatrix}\delta x \\ \delta u\end{bmatrix} dt$$

which is subject to

$$\begin{aligned} \delta\dot{x} &= f_x \delta x + f_u \delta u \\ \delta\psi_f &= \psi_x(x(t_f))\delta x(t_f) \end{aligned}$$

where $\delta x(t_0)$ and $\delta\phi_f$ are given. Then if we let $A(t) = f_x$ and $B(t) = f_u$ we have

$$\delta\dot{x} = A(t)\delta x + B(t)\delta u.$$

Also, let $R_1 = H_{xx}$, $R_{12} = R_{21}^T = H_{xu}$, $R_{21} = H_{ux}$, and $R_2 = H_{uu}$ which means we have

$$\delta^2 J_a = \frac{1}{2}\delta x^T(t_f)\phi_{xx}(t_f)\delta x(t_f) + \frac{1}{2}\int_0^{t_f} \begin{bmatrix}\delta x^T & \delta u^T\end{bmatrix}\begin{bmatrix}R_1 & R_{12}\\ R_{12}^T & R_2\end{bmatrix}\begin{bmatrix}\delta x\\ \delta u\end{bmatrix}dt$$

The Hamiltonian for this accessory minimization problem becomes

$$H_{acc} = \frac{1}{2}\begin{bmatrix}\delta x^T & \delta u^T\end{bmatrix}\begin{bmatrix}R_1 & R_{12}\\ R_{12}^T & R_2\end{bmatrix}\begin{bmatrix}\delta x\\ \delta u\end{bmatrix} + \delta\lambda(A\delta x + B\delta u).$$

The costate equation gives

$$\delta\dot{\lambda} = -\frac{\partial H_{acc}}{\partial x} = -R_1\delta x - R_{12}\delta u - A^T\delta\lambda.$$

The control becomes

$$\begin{aligned}\frac{\partial H_{acc}}{\partial u} &= R_{12}^T\delta x + R_2\delta u + B^T\delta\lambda = 0\\ \therefore \delta u &= -R_2^{-1}\left(R_{12}^T\delta x + B^T\delta\lambda\right)\\ \left(\delta u \right.&= \left.-H_{uu}^{-1}\left(H_{ux}\delta x + f_u^T\delta\lambda\right)\right)\end{aligned}$$

when $R_2 = H_{uu}$ is nonsingular. Now if we plug the open loop control back into the dynamics we have

$$\begin{aligned}\delta\dot{x} &= (A - BR_2^{-1}R_{12}^T)\delta x - BR_2^{-1}B^T\delta\lambda\\ \delta\dot{\lambda} &= (R_{12}R_2^{-1}R_{12}^T - R_1)\delta x - (A^T - R_{12}R_2^{-1}B^T)\delta\lambda.\end{aligned}$$

Rewriting this in the state space form we get

$$\begin{bmatrix}\delta\dot{x}\\ \delta\dot{\lambda}\end{bmatrix} = \begin{bmatrix}\tilde{A}(t) & -\Sigma(t)\\ -\tilde{R}(t) & -\tilde{A}^T(t)\end{bmatrix}\begin{bmatrix}\delta x\\ \delta\lambda\end{bmatrix}$$

where

$$\tilde{A} = A - BR_2^{-1}R_{12}^T, \quad \Sigma = BR_2^{-1}B^T, \quad \tilde{R} = R_{12}R_2^{-1}R_{12}^T - R_1.$$

The boundary conditions being

$$\begin{aligned}\delta\lambda(t_f) &= \frac{\partial}{\partial x}\left[\frac{1}{2}\delta x^T\phi_{xx}\delta x + \lambda v(\psi_x(x(t_f))\delta x - \delta\psi_f)\right]\\ \delta\lambda(t_f) &= \psi_{xx}(x(t_f))\delta x(t_f) + \psi_x^T(x(t_f))dv.\end{aligned}$$

Then, assuming

$$\begin{aligned}\delta\lambda(t) &= P(t)\delta x(t) + S(t)dv\\ \delta\psi_f &= S^T(t)\delta x(t) + Q(t)dv\end{aligned}$$

where

$$P(t_f) = \phi_{xx}(x(t_f)), \quad S(t_f) = \psi_x^T(x(t_f)), \quad Q(t_f) = 0.$$

Then taking the derivative of the two equations above we have

$$\begin{aligned} \delta\dot{\lambda} &= \dot{P}\delta x + P\delta\dot{x} + \dot{S}dv \\ 0 &= \dot{S}^T\delta x + S^T\delta\dot{x} + \dot{Q}dv \end{aligned}$$

We can equate the first equation with

$$\begin{bmatrix} \delta\dot{x} \\ \delta\dot{\lambda} \end{bmatrix} = \begin{bmatrix} \tilde{A}(t) & -\Sigma(t) \\ -\tilde{R}(t) & -\tilde{A}^T(t) \end{bmatrix} \begin{bmatrix} \delta x \\ \delta\lambda \end{bmatrix}$$

and we get

$$\begin{gathered} -\tilde{R}\delta x - \tilde{A}^T\delta\lambda = \dot{P}\delta x + P(\tilde{A}\delta x - \Sigma\delta\lambda) + \dot{S}dv \\ (-\tilde{R} - \tilde{A}^TP - \dot{P} - P\tilde{A} + P\Sigma P)\delta x - (\tilde{A}^TS - P\Sigma S + \dot{S})dv = 0. \end{gathered}$$

Thus, we have the following relations

$$\begin{aligned} -\dot{P} &= P\tilde{A} + \tilde{A}^TP + \tilde{R} - P\Sigma P \\ \dot{S} &= -(\tilde{A}^T - P\Sigma)S \\ \dot{Q} &= S^T\Sigma S \end{aligned}$$

where we have

$$\begin{aligned} P(t_f) &= \phi_{xx}(x(t_f)) \\ S(t_f) &= \psi_x^T(x(t_f)) \\ Q(t_f) &= 0. \end{aligned}$$

Here we can deduce that if $\delta x(t_0)$ is known, $\delta\psi_f$ is also known

$$dv = Q^{-1}(t_0)\left[\delta\psi_f - S^T(t_f)\delta x(t_0)\right].$$

Finally we can compute the optimal control for the Neighboring Optimal Guidance (NOG) or Neighboring Optimal Control (NOC) to be

$$\begin{aligned} \delta u &= -R_2^{-1}R_{12}^T\delta x - R_2^{-1}B^T\delta\lambda \\ &= -R_2^{-1}R_{12}^T\delta x - R_2^{-1}B^T\left[(P - SQ^{-1}S^T)\delta x + SQ^{-1}\delta\psi_f\right] \end{aligned}$$

$$\boxed{\delta u = -\Gamma_1(t)\delta x - \Gamma_2(t)\delta\psi_f.}$$

where

$$\Gamma_1 = R_2^{-1}(R_{12}^T + B^T(P - SQ^{-1}S^T))$$
$$\Gamma_2 = R_2^{-1}B^T SQ^{-1}.$$

Now taking the Figure 1. of a paper by Hui Yan et al. as an example we can construct a block diagram of the neighboring guidance scheme [2].

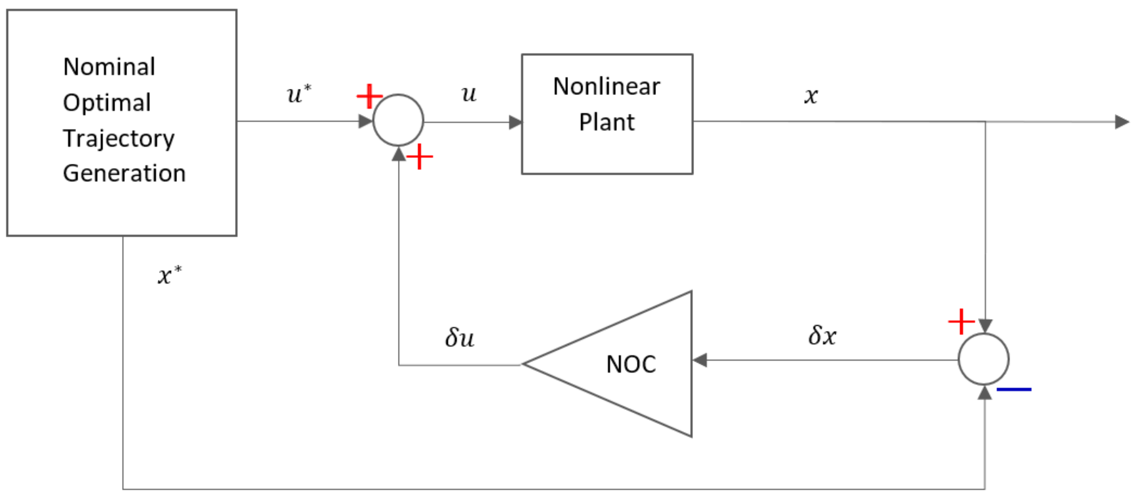


Figure 9: Neighboring Optimal Guidance (NOG) Diagram (NOC: Neighboring Optimal Control)

(e) For a vertical wind disturbance, $W$ modelled as force (N), we can manipulate the dynamics to be the following

$$\dot{r} = v \sin\gamma$$
$$\dot{\phi} = \frac{v}{r}\cos\gamma$$
$$\dot{v} = -\frac{D}{m} - \frac{\mu \sin\gamma}{r^2}$$
$$\dot{\gamma} = \frac{L}{mv} - \frac{\mu\cos\gamma}{r^2 v} + \frac{v\cos\gamma}{r} + \frac{|W|}{mv}\,\mathrm{sgn}(W)$$

This is because the vertical component is assumed to only effect the flight path angle, $\gamma$. This dynamic model will represent the nonlinear plant shown on Figure 9. However, this plant will only be active at $t = 1000$ otherwise the original plant of

$$
\begin{aligned}
\dot{r} &= v \sin\gamma \\
\dot{\phi} &= \frac{v}{r}\cos\gamma \\
\dot{v} &= -\frac{D}{m} - \frac{\mu \sin\gamma}{r^2} \\
\dot{\gamma} &= \frac{L}{mv} - \frac{\mu \cos\gamma}{r^2 v} + \frac{v \cos\gamma}{r}
\end{aligned}
$$

will be implemented. Next, we will have to compute the NOC.

## References

[1] M. Athans, P. L. Falb, and R. T. Lacoss, “Optimal control of self-adjoint systems,” *IEEE Transactions on Applications and Industry*, vol. 83, no. 72, pp. 161–166, 1964.

[2] H. Yan, I. M. Ross, and F. Fahroo, “Real-time computation of neighboring optimal control laws,” 2002.

# Appendix

## 9.1 Problem 4: MATLAB Code

```
% AE6511 Final Problem 4
% Tomoki Koike
clear all; close all; clc;  % housekeeping commands
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%%
syms x_1(t) x_2(t) u
DE1 = diff(x_1,t) == x_2 + u;
DE2 = diff(x_2,t) == -u;

% (A) When u = -1 and IC applied
IC = [x_1(0) == 0, x_2(0) == -0.5];
DE1A = subs(DE1,u,-1);
DE2A = subs(DE2,u,-1);
[bb1_x1(t), bb1_x2(t)] = dsolve([DE1A DE2A],IC);
expand(bb1_x1)
bb1_x2
%%
% (B) When u = 1 and IC applied
DE1B = subs(DE1,u,1);
DE2B = subs(DE2,u,1);
[bb2_x1(t), bb2_x2(t)] = dsolve([DE1B DE2B],IC);
expand(bb2_x1)
bb2_x2
%%
% Optimal trajectories
tspan = linspace(0,4,100);
x2 = linspace(-2,2,100);

% Singular arcs
x1s = -2*x2;
y1s = zeros(size(x2));

fig = figure("Renderer","painters","Position",[60 60 950 800]);
    plot(bb1_x1(tspan),bb1_x2(tspan),'DisplayName','(1)')
    hold on;
    plot(bb2_x1(tspan),bb2_x2(tspan),'DisplayName','(2)')
    plot(y1s, x2,'DisplayName','(3)')
    plot(x1s, x2,'DisplayName','(4)')
```

```
    plot(0,-0.5,'.k','MarkerSize',15,'DisplayName','$t_0$')
    plot(0, 0, '.r', 'MarkerSize', 15,'DisplayName','$T$')
    grid on; grid minor; box on; hold off;
    legend()
    xlabel('$x_1$')
    ylabel('$x_2$')
    xlim([-2,1])
    ylim([-2,2])
saveas(fig,'outputs/p4_all_traj.png')

%%
% Find the optimal trajectory
syms y_1 y_2
eqn1 = y_1 == 0.5*(y_2+0.5)^2 - 1.5*(y_2+0.5);
eqn2 = y_2 == -0.5*y_1;
sol = solve([eqn1 eqn2], [y_1 y_2])
sol.y_1
sol.y_2
%%
% Appropriate intersect
insct.x1 = sol.y_1(1);
insct.x2 = sol.y_2(1);

% Compute the time to the intersect
insct.t = solve([bb1_x1(t)==insct.x1 bb1_x2(t)==insct.x2], t)
%%
% singular
dX = [diff(x_1,t); diff(x_2,t)];
X = [x_1; x_2];
A = [-1 0; 1 1];
cond = [x_1(1)==-1, x_2(1)==0.5];
sig_traj = dsolve(dX == A*X,cond)
sig_traj.x_1
sig_traj.x_2
%%
% (b)
% u = 1 bang-bang arc through origin
syms c_1 c_2 tau
assume(tau, 'positive');
eqn3 = -tau + c_1 == tau - 0.5;
eqn4 = -0.5*tau^2 + (c_1 + 1)*tau + c_2 == 0.5*tau^2 - 1.5*tau;
eqn5 = 0 == -0.5 + 0.5*(c_1 + 1)^2 + c_2;
[c1, c2, tt] = solve([eqn3 eqn4 eqn5], [c_1 c_2 tau]);
c1 = double(c1)
c2 = double(c2)
```

```
tt = double(tt)
%%
% New boundary
bb3_x1(t) = -0.5*t^2 + (c1 + 1)*t + c2;
bb3_x2(t) = -t + c1;

tspan1 = linspace(tt,4.5,100);
tspan = linspace(0,4,100);
%%
fig = figure("Renderer","painters","Position",[60 60 950 800]);
    plot(bb1_x1(tspan),bb1_x2(tspan),'DisplayName','(1)')
    hold on;
    plot(bb2_x1(tspan),bb2_x2(tspan),'DisplayName','(2)')
    plot(y1s, x2,'DisplayName','(3)')
    plot(x1s, x2,'DisplayName','(4)')
    plot(bb3_x1(tspan1),bb3_x2(tspan1),'DisplayName','(5)')
    plot(0,-0.5,'.k','MarkerSize',15,'DisplayName','$t_0$')
    plot(0, 0, '.r', 'MarkerSize', 15,'DisplayName','$T$')
    plot(bb3_x1(tt),bb3_x2(tt),'.','MarkerSize',15,'DisplayName','$\tau$')
    grid on; grid minor; box on; hold off;
    legend()
    xlabel('$x_1$')
    ylabel('$x_2$')
    xlim([-2,1])
    ylim([-2,2])
saveas(fig,'outputs/p4_all_traj_new.png')
%%
% (c)
tau01 = solve([(-0.5*t^2 + (c1 + 1)*t + c2)==0],t);
tau02 = solve((-t + c1)==0,t);
tt0 = double(mean([tau01(1) tau02]))
%%
xopt1_x1 = @(t) 0.5 * (0.5*t.^2 - 1.5.*t).^2;
xopt2_x1 = @(t) 0.5 * (-2.7183.*exp(-t)).^2;
Jopt = integral(xopt1_x1,0,double(insct.t)) + integral(xopt2_x1,double(insct
    .t),inf)

xbb1_x1 = xopt1_x1;
xbb2_x1 = @(t) 0.5 * (-0.5*t.^2 + (c1 + 1)*t + c2).^2;
Jbb = integral(xbb1_x1,0,tt) + integral(xbb2_x1,tt,tt0)
```

### 9.2 Problem 6: MATLAB Code

```
% AE6511 Final Problem 6
% Tomoki Koike
clear all; close all; clc;  % housekeeping commands
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%%
% BVP4C
x0 = [0.1361 0.8693 0.5797 0.5499 0.1450 0.8530 0.6221 0.3510 0.5132]
% x0 = rand(1,9);
mesh = linspace(0, 1, 10);
solinit = bvpinit(mesh, x0);
opts = bvpset('RelTol',1e-5, 'AbsTol',1e-6,'Stats','on','Nmax',10000);
sol = bvp4c(@odefcn,@bcfcn,solinit, opts);
%%
% Unpack results
T = linspace(0,1,1000);
[xopt,xdopt] = deval(sol,T);
xopt = xopt.'; xdopt = xdopt.';
tf_eval = mean(xopt(1,9));
tf_out = abs(tf_eval);
tspan = tf_out * T;

x1_sol = xopt(:,1);
x2_sol = xopt(:,2);
x3_sol = xopt(:,3);
x4_sol = xopt(:,4);
lambda1_sol = xopt(:,5);
lambda2_sol = xopt(:,6);
lambda3_sol = xopt(:,7);
lambda4_sol = xopt(:,8);

u1 = -lambda2_sol;
u2 = -lambda4_sol;

H = (0.5*(u1.^2 + u2.^2) + lambda1_sol.*xdopt(:,1)/tf_eval ...
    + lambda2_sol.*xdopt(:,2)/tf_eval ...
    + lambda3_sol.*xdopt(:,3)/tf_eval ...
    + lambda4_sol.*xdopt(:,4)/tf_eval);
H = round(H,4);
%%
% J = 0.5*(x2_sol(end)-2)^2 + 0.5*(trapz(u1.^2) + trapz(u2.^2));

```

```
% Compute the controls (linear) from data
b1 = u1(1);
a1 = (u1(end)-b1) / tf_out;
b2 = u2(1);
a2 = (u2(end)-b2) / tf_out;
f = @(t) 0.5*((a1*t + b1).^2 + (a2*t + b2).^2);

% The minimized objective
J_min = 0.5*(x2_sol(end)-2)^2 + integral(f,0,tf_out)
%%
fig = figure("Renderer","painters","Position",[60 60 900 800]);
    % x vs t
    subplot(2,2,1)
    plot(tspan, x1_sol,'DisplayName','$x_1$')
    hold on;
    plot(tspan, x2_sol,'DisplayName','$x_2$')
    plot(tspan, x3_sol,'DisplayName','$x_3$')
    plot(tspan, x4_sol,'DisplayName','$x_4$')
    xlabel('$t$')
    ylabel('$x$')
    grid on; grid minor; box on; hold off; legend('Location','best');
    % lambda vs t
    subplot(2,2,2)
    plot(tspan, lambda1_sol,'DisplayName','$\lambda_1$')
    hold on;
    plot(tspan, lambda2_sol,'DisplayName','$\lambda_2$')
    plot(tspan, lambda3_sol,'DisplayName','$\lambda_3$')
    plot(tspan, lambda4_sol,'DisplayName','$\lambda_4$')
    xlabel('$t$')
    ylabel('$\lambda$')
    grid on; grid minor; box on; hold off; legend('Location','best');
    % u vs t
    subplot(2,2,3)
    plot(tspan, u1,'DisplayName','$u_1$')
    hold on;
    plot(tspan, u2,'DisplayName','$u_2$')
    xlabel('$t$')
    ylabel('$u$')
    grid on; grid minor; box on; hold off; legend('Location','best');
    % Hamiltonian
    subplot(2,2,4)
    plot(tspan, H)
    xlabel('$t$')
    ylabel('$H$')
    ytickformat('%,.2f')
```

```
    grid on; grid minor; box on;
% saveas(fig, 'p4c.png');
%% Function

function dxdt = odefcn(t,x)
    dxdt = zeros(9,1);
    dxdt(1) = x(2); % x1
    dxdt(2) = -x(6); % x2
    dxdt(3) = x(4); % x3
    dxdt(4) = -x(8); % x4
    dxdt(5) = 0;  % lambda1
    dxdt(6) = -x(5);  % lambda2
    dxdt(7) = 0;  % lambda3
    dxdt(8) = -x(7);  % lambda4
    dxdt(9) = 0; % tf
    dxdt = dxdt * x(9);
end

function res = bcfcn(xa,xb)
    res = zeros(9,1);
    res(1) = xa(1) - 1;  % x1(0)
    res(2) = xa(2);  % x2(0)
    res(3) = xa(3) + 1;  % x3(0)
    res(4) = xa(4);  % x4(0)
    res(5) = xb(1) - xb(3);  % x1(tf) - x3(tf)
    res(6) = xb(2) - xb(4);  % x2(tf) - x4(tf)
    res(7) = xb(5) + xb(7);  % lambda1(tf) + lambda3(tf)
    res(8) = xb(6) + xb(8) - xb(2) + 2;  % lambda2(tf) + lambda4(tf) - x2(tf
       ) + 2
    res(9) = (-0.5*(xb(6)^2 + xb(8)^2) + xb(5)*xb(2) - xb(6)*xb(6) + ...
        xb(7)*xb(4) - xb(8)*-xb(8)) * xb(9); % H(t_f)
end
```

### 9.3 Problem 7.1: MATLAB Code

```
% AE6511 Final Problem 7
% Tomoki Koike
clear all; close all; clc;  % housekeeping commands
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%% Settings

```

```
% State Parameters
auxdata.m = 38000;            % [kg]
auxdata.mu = 3.986e14;        % [m^3/s^2]
auxdata.rho0 = 1.225;         % [kg/m^3]
auxdata.Hs = 7254.24;         % [m]
auxdata.r0 = 6371e3;          % [m]

% State Boundary Conditions
h0 = 121.9e3;                 % [m]
hf = 30.48e3;                 % [m]
phi0 = deg2rad(-25);          % [rad]
phif_min = 0;                 % [rad]
phif_max = deg2rad(1200);     % [rad]
v0 = 7626;                    % [m/s]
vf = 908.15;                  % [m/s]
gamma0 = -deg2rad(-1.25);     % [rad]
gammaf_min = -deg2rad(6);     % [rad]
gammaf_max = 0;               % [rad]

t0 = 0;                       % [s]
tf_min = 100;                 % [s]
tf_max = 86400;               % [s]

% Control Bounds
alpha_min = -5;               % [deg]
alpha_max = 40;               % [deg]

% State Bounds for Optimization
h_bd = [0 200e3];                       % [m]
phi_bd = [phi0 phif_max];               % [rad]
v_bd = [400 10000];                     % [m/s]
gamma_bd = [-deg2rad(60) deg2rad(60)];  % [rad]

% Aircraft Characteristics
auxdata.S = 149.3881;                   % [m^2]
Lambda_max = 4e5;                       % [W/m^2]
q_max = 14500;                          % [kg/ms^2]
g0 = 9.81;                              % [m/s^2]
n_max = 5*g0;                           % [m/s^2]
auxdata.k_lambda = 9.4369e-5;           % [kg^0.5/m^1.5]

% Lift/Drag -> [deg]
auxdata.a0 = -0.20704;
auxdata.a1 = 0.029244;
auxdata.b0 = 0.07854;
```

```
auxdata.b1 = -0.61592e-2;
auxdata.b2 = 0.621408e-3;
%% Define bounds

% Time
bounds.phase.initialtime.lower = t0;
bounds.phase.initialtime.upper = t0;
bounds.phase.finaltime.lower = tf_min;
bounds.phase.finaltime.upper = tf_max;

% Initial states
bounds.phase.initialstate.lower = [auxdata.r0+h0 phi0 v0 gamma0];
bounds.phase.initialstate.upper = [auxdata.r0+h0 phi0 v0 gamma0];

% Midway states
bounds.phase.state.lower = [auxdata.r0+h_bd(1) phi_bd(1) v_bd(1) gamma_bd(1)
    ];
bounds.phase.state.upper = [auxdata.r0+h_bd(2) phi_bd(2) v_bd(2) gamma_bd(2)
    ];

% Final states
bounds.phase.finalstate.lower = [auxdata.r0+hf phif_min vf gammaf_min];
bounds.phase.finalstate.upper = [auxdata.r0+hf phif_max vf gammaf_max];

% Controls
bounds.phase.control.lower = [alpha_min];
bounds.phase.control.upper = [alpha_max];

% Path
bounds.phase.path.lower = [0 0 0];
bounds.phase.path.upper = [Lambda_max q_max n_max];

%% Guess

phif_guess = pi/6;
gammaf_guess = -deg2rad(2);
guess.phase.time = [t0; tf_max];
guess.phase.state(:,1) = [auxdata.r0+h0; auxdata.r0+hf];
guess.phase.state(:,2) = [phi0; phif_guess];
guess.phase.state(:,3) = [v0; vf];
guess.phase.state(:,4) = [gamma0; gammaf_guess];
guess.phase.control = [30; 30];
%% Mesh

mesh.method = 'hp-LiuRao-Legendre';
```

```
mesh.tolerance = 1e-6;
mesh.maxiterations = 8;
mesh.colpointsmin = 4;
mesh.colpointsmax = 20;
mesh.sigma = 0.75;
%% Solver

% Configure setup using the infromation provided
setup.name= 'HSvehicle-system'
setup.functions.continuous = @HSvehicle_systemContinuous;
setup.functions.endpoint = @HSvehicle_systemEndpoint;
setup.displaylevel = 2;
setup.bounds = bounds;
setup.guess = guess;
setup.auxdata = auxdata;
setup.mesh = mesh;
setup.nlp.solver = 'ipopt';
setup.derivatives.supplier = 'sparseCD';
setup.derivatives.derivativelevel = 'second';
setup.method = 'RPM-Differentiation';
%% Solve problem

output = gpops2(setup);
%% Plot Results

t = output.result.solution.phase.time;                   % time
x1 = output.result.solution.phase.state(:,1);            % r
x2 = output.result.solution.phase.state(:,2);            % phi
x3 = output.result.solution.phase.state(:,3);            % v
x4 = output.result.solution.phase.state(:,4);            % gamma
u = output.result.solution.phase.control;                % alpha
lambda1 = output.result.solution.phase.costate(:,1);    % costate 1
lambda2 = output.result.solution.phase.costate(:,2);    % costate 2
lambda3 = output.result.solution.phase.costate(:,3);    % costate 3
lambda4 = output.result.solution.phase.costate(:,4);    % costate 4

% Compute Hamiltonian
rho = auxdata.rho0*exp(-(x1 - auxdata.r0)/auxdata.Hs);
CD = auxdata.b0 + auxdata.b1*u + auxdata.b2*u.^2;
CL = auxdata.a0 + auxdata.a1*u;
D = 0.5*rho.*x3.^2*auxdata.S.*CD;
L = 0.5*rho.*x3.^2*auxdata.S.*CL;
m = auxdata.m;
mu = auxdata.mu;
```

```
H = (lambda1.*(x3.*sin(x4)) ...
    + lambda2.*(x3.*cos(x4)./x1) ...
    + lambda3.*(-D/m - mu*sin(x4)./x1.^2) ...
    + lambda4.*(L/m./x3 - mu*cos(x4)./x1.^2./x3 + x3.*cos(x4)./x1));
%%
% Trajectories
fig = figure("Renderer","painters","Position",[60 60 1050 950]);
    subplot(2,2,1)
    plot(t,x1-auxdata.r0,'-','Color','#0320fc')
    grid on; grid minor; box on;
    xlabel('$t$ [s]')
    ylabel('$h$ [m]')
    subplot(2,2,2)
    plot(t,x2,'-','Color','#1ecc0e')
    grid on; grid minor; box on;
    xlabel('$t$ [s]')
    ylabel('$\phi$ [rad]')
    subplot(2,2,3)
    plot(t,x3,'-','Color','#e6be20')
    grid on; grid minor; box on;
    xlabel('$t$ [s]')
    ylabel('$v$ [m/s]')
    subplot(2,2,4)
    plot(t,x4,'-','Color','#fc0303')
    grid on; grid minor; box on;
    xlabel('$t$ [s]')
    ylabel('$\gamma$ [rad]')
saveas(fig,'outputs/p7_optStates.png')
%%
% Control
fig = figure("Renderer","painters","Position",[60 60 950 800]);
    plot(t,u)
    xlabel('$t$ [s]')
    ylabel('$\alpha$ [deg]')
    grid on; grid minor; box on;
saveas(fig,'outputs/p7_optControl.png')
%%
% Costates
fig = figure("Renderer","painters","Position",[60 60 950 950]);
    subplot(2,2,1)
    plot(t,lambda1,'-','Color','#0320fc')
    grid on; grid minor; box on;
    xlabel('$t$ [s]')
    ylabel('$\lambda_1$')
    subplot(2,2,2)
```

```matlab
    plot(t,lambda2,'-','Color','#1ecc0e')
    grid on; grid minor; box on;
    xlabel('$t$ [s]')
    ylabel('$\lambda_2$')
    subplot(2,2,3)
    plot(t,lambda3,'-','Color','#e6be20')
    grid on; grid minor; box on;
    xlabel('$t$ [s]')
    ylabel('$\lambda_3$')
    subplot(2,2,4)
    plot(t,x4,'-','Color','#fc0303')
    grid on; grid minor; box on;
    xlabel('$t$ [s]')
    ylabel('$\lambda_4$')
saveas(fig,'outputs/p7_optStates.png')
%%
% Hamiltonian
fig = figure("Renderer","painters","Position",[60 60 950 800]);
    plot(t,H)
    xlabel('$t$ [s]')
    ylabel('$H$')
    grid on; grid minor; box on;
saveas(fig,'outputs/p7_optHamiltonian.png')
%% Dynamics and Objectives

function phaseout = HSvehicle_systemContinuous(input)
    r = input.phase.state(:,1);
    phi = input.phase.state(:,2);
    v = input.phase.state(:,3);
    gamma = input.phase.state(:,4);
    alpha = input.phase.control(:,1);  % [deg]

    % Parameters
    mu = input.auxdata.mu;
    m = input.auxdata.m;
    S = input.auxdata.S;
    rho0 = input.auxdata.rho0;
    r0 = input.auxdata.r0;
    Hs = input.auxdata.Hs;
    a0 = input.auxdata.a0;
    a1 = input.auxdata.a1;
    b0 = input.auxdata.b0;
    b1 = input.auxdata.b1;
    b2 = input.auxdata.b2;
    k_lambda = input.auxdata.k_lambda;
```

```
    % Air density
    rho = rho0.*exp(-(r-r0)./Hs);

    % Lift/Drag
    CD = b0 + b1.*alpha + b2.*alpha.^2;
    CL = a0 + a1.*alpha;
    L = 0.5*rho.*v.^2*S.*CL;
    D = 0.5*rho.*v.^2*S.*CD;

    % Dynamics
    dr = v.*sin(gamma);
    dphi = v.*cos(gamma)./r;
    dv = -D/m - mu*sin(gamma)./r.^2;
    dgamma = L/m./v - mu*cos(gamma)./r.^2./v + v.*cos(gamma)./r;

    % Path
    Lambda = k_lambda*sqrt(rho).*v.^3;  % heating
    q = 0.5*rho.*v.^2;                  % dynamic pressure
    n = sqrt(L.^2 + D.^2)/m;            % normal load

    phaseout.path = [Lambda q n];
    phaseout.dynamics = [dr dphi dv dgamma];
end

function output = HSvehicle_systemEndpoint(input)
    % Maximize phi(t_f) == Minimize -phi(t_f)
    output.objective = -input.phase.finalstate(2);
end
```