College of Engineering
School of Aerospace Engineering

AE 6511: Optimal Guidance and Controls

HW6

Professor:
Panagiotis Tsiotras
Gtech AE Professor

Student:
Tomoki Koike
Gtech MS Student

November 19, 2021

Table of Contents

1 Problem 1 **2**

2 Problem 2 **3**

3 Problem 3 **10**

4 Problem 4 **12**

5 Problem 5 **15**

6 Problem 6 **20**

7 Problem 7 **24**

8 Problem 8 **26**

9 Appendix **28**

9.1 Problem 2a: MATLAB Code 28

9.2 Problem 2b: MATLAB Code 30

9.3 Problem 2c: MATLAB Code 33

9.4 Problem 5: MATLAB Code 36

9.5 Problem 6: MATLAB Code 39

9.6 Problem 7: MATLAB Code 41

Problem 1

Study the material of Chapter 7 (Linear−Quadratic Optimal Control) posted on Canvas.

Problem 2

Write a MATLAB code based on the document uploaded to Canvas to solve the following problem. A ship is located at the point $(x_0, y_0) = (-20, 0)$ml at time $t = 0$ when it encounters a medical emergency and it has to reach the shore as soon as possible. It is known that there is a small city at the location $(x_1, y_1) = (-15, 35.5)$ml with a medical center. As the captain of the ship, you are to determine the fastest possible route to the city. It is assumed that the speed of the ship with respect to the water is constant, $v = 15ml/hr$. You also know the speed and direction of the sea currents in the area, which are given to you from a meteorological satellite as $\vec{v}_c = u(x, y)\hat{\mathbf{i}} + v(x, y)\hat{\mathbf{j}}$.

(a) Derive the necessary conditions for the optimal control strategy, and calculate the optimal path and the time to reach the city, assuming that the currents are constant, given by

$$\vec{v}_c = 2\hat{\mathbf{i}} - 6\hat{\mathbf{j}}$$

Plot the optimal path in the $x - y$ plane along with the vectors showing the direction of the currents.

(b) When you are about to start your dash to the shore, you learn that the doctor in the medical center will be able to fly by helicopter to any point at the shore to pick up the patient. Find the new optimal path and the time to reach the shore, assuming that the contour of the shoreline is known to be

$$\Psi(x, y) = 25 - 0.25x - 0.002x^3 - y = 0.$$

Plot the optimal path in the $x - y$ plane along with the vectors showing the direction of the currents.

(c) An update of the meteorological data from the satellite shows that strong winds have developed in the area and that the currents have changed significantly. The new currents are

$$\vec{v}_c = -(y - 50)\hat{\mathbf{i}} + 2(x - 15)\hat{\mathbf{j}}.$$

Recalculate the optimal control and plot the optimal path in the $x - y$ plane along with the vectors showing the direction of the currents. Plot the optimal steering angle history $\theta^*(t)$.

In call cases, plot the Hamiltonian and verify that it remains zero for all time.

Solution:
The differential equations for this system can be formulated as follows from the given statement.

$$\dot{x} = V\cos\theta + u(x,y)$$
$$\dot{y} = V\sin\theta + v(x,y)$$

Let the performance index and the Hamiltonian for this problem be

$$J = t_f$$
$$H = \lambda_1(V\cos\theta + u) + \lambda_2(V\sin\theta + v).$$

For this problem the costate equations become

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x} = -\lambda_1 u_x - \lambda_2 v_x$$
$$\dot{\lambda}_2 = -\frac{\partial H}{\partial x} = -\lambda_1 u_y - \lambda_2 v_y$$

and the Hamiltonian does not depend on time; therefore,

$$\dot{H} = 0 \to \tan\theta = \frac{\lambda_2}{\lambda_1}$$
$$H(t_f) = -1;$$

From this we can compute the relations

$$\lambda_1 = -\frac{\cos\theta}{V + u\cos\theta + v\sin\theta}$$

$$\lambda_2 = -\frac{\sin\theta}{V + u\cos\theta + v\sin\theta}$$

$$\dot{\theta} = \sin^2\theta u_x + \sin\theta\cos\theta(u_x - v_y) - \cos^2\theta u_y.$$

These are the necessary conditions for this problem. Using the boundary conditions we are able to solve this problem analytically or numerically.

(a) With the constant current of $\vec{v}_c = 2\hat{\mathbf{i}} - 6\hat{\mathbf{j}}$ we have

$$\dot{x} = V\cos\theta + 2$$
$$\dot{y} = V\sin\theta - 6$$

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x} = 0$$
$$\dot{\lambda}_2 = -\frac{\partial H}{\partial x} = 0$$

Referring to the code in ?? we obtain the following results. From Figure 1 we can observe that the Hamiltonian is fluctuating at an infinitesimal value, and therefore can be approximated to a constant 0.

$$\boxed{\min t_f = 1.7578.}$$

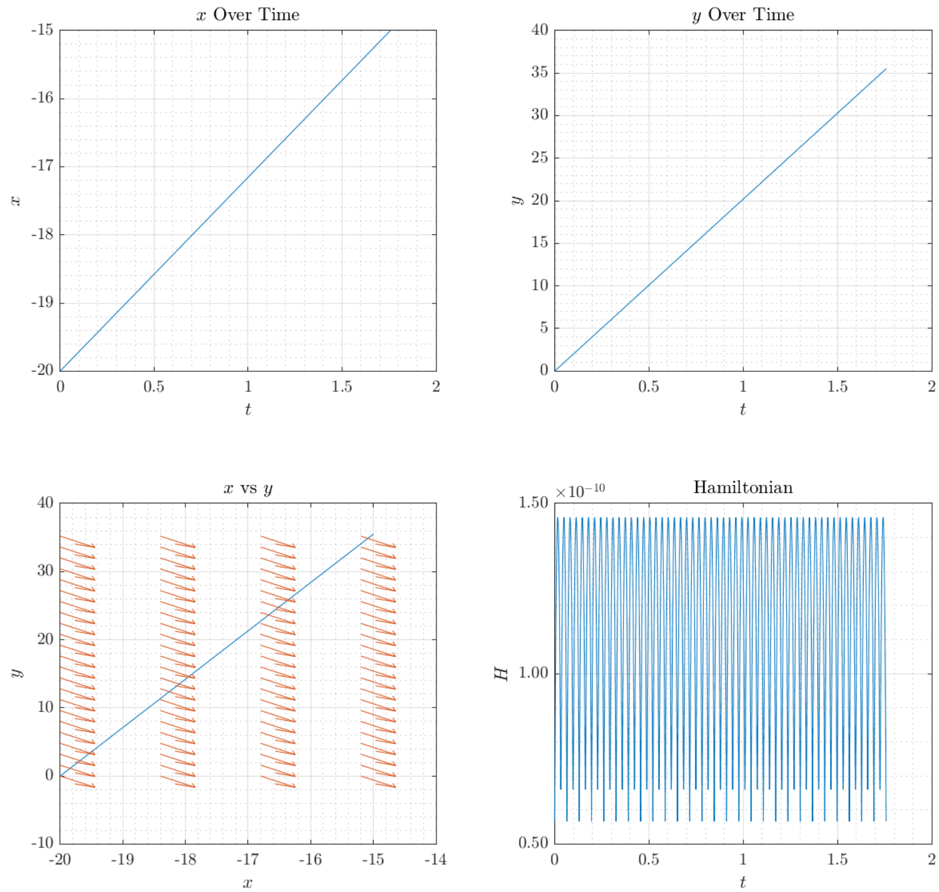

Figure 1: Problem 4(a) optimal trajectories and the Hamiltonian

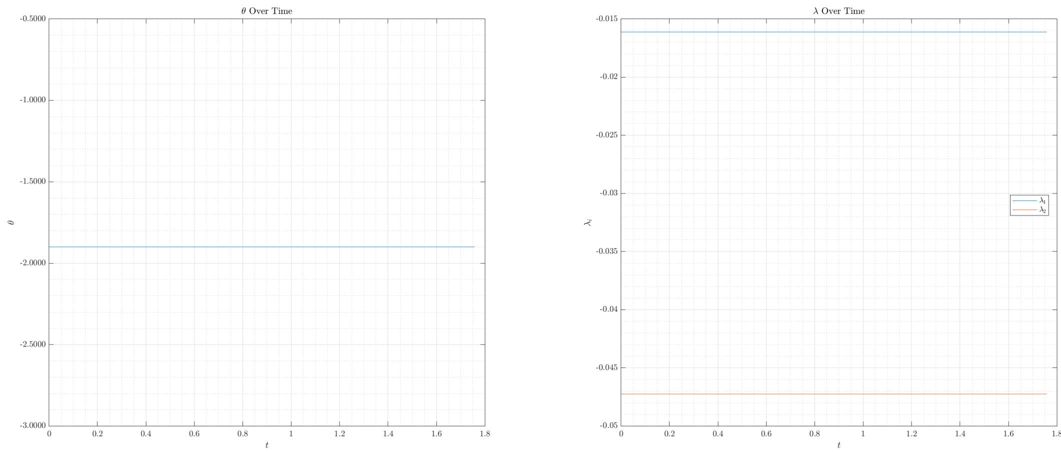

Figure 2: Problem 4(a) optimal control and costate

(b) For this case we have a terminal constraint of

$$\Psi(t_f, x(t_f)) = 25 - 0.25x_f - 0.002x_f^3 - y_f = 0.$$

and a terminal cost of

$$\Phi(t_f, x(t_f)) = (x_f + 15)^2 + (y_f - 35.5)^2$$

Only the transversality condition changes from problem (a), which becomes

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

$$\begin{bmatrix} H(t_f) \\ -\lambda_1(t_f) + 2(x_f + 15) \\ -\lambda_2(t_f) + 2(y_f - 35.5) \end{bmatrix} = \begin{bmatrix} 0 \\ -0.25 - 0.006x_f^2 \\ -1 \end{bmatrix} \zeta$$

$$\text{where } \zeta = \text{ const.}$$

Thus, we have

$$-\lambda_1(t_f) + 2(x_f + 15) = (-0.25 - 0.006x_f^2)(\lambda_2(t_f) - 2(y_f - 35.5)).$$

and we know that

$$25 - 0.25x_f - 0.002x_f^3 - y = 0.$$

Now we can solve the problem numerically. This gave us the result of

$$\boxed{t_f = 3.6181.}$$

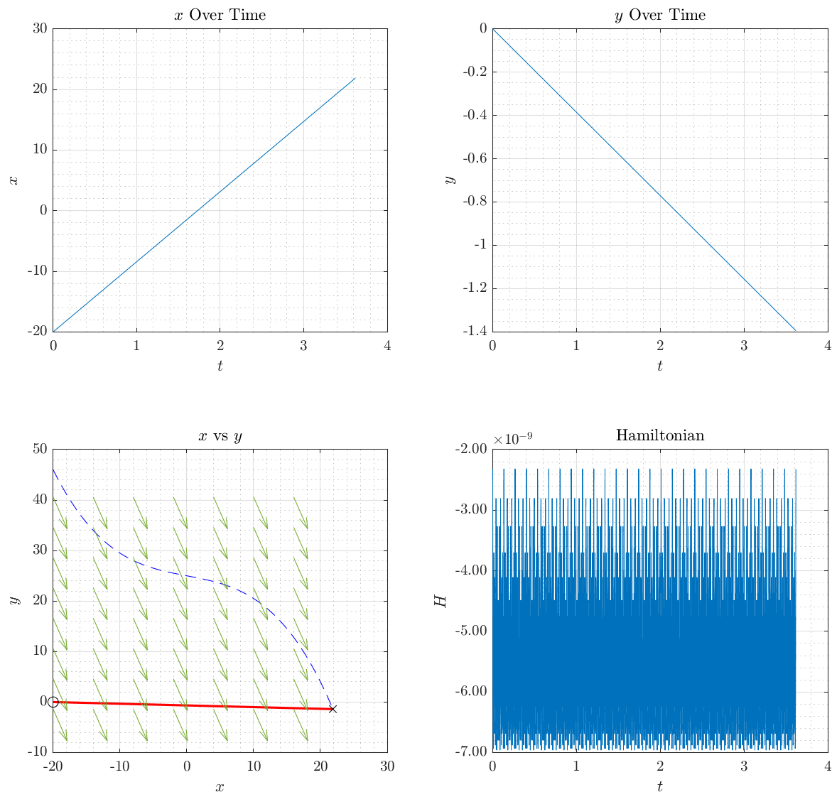

Figure 3: Problem 4(b) optimal trajectories and the Hamiltonian

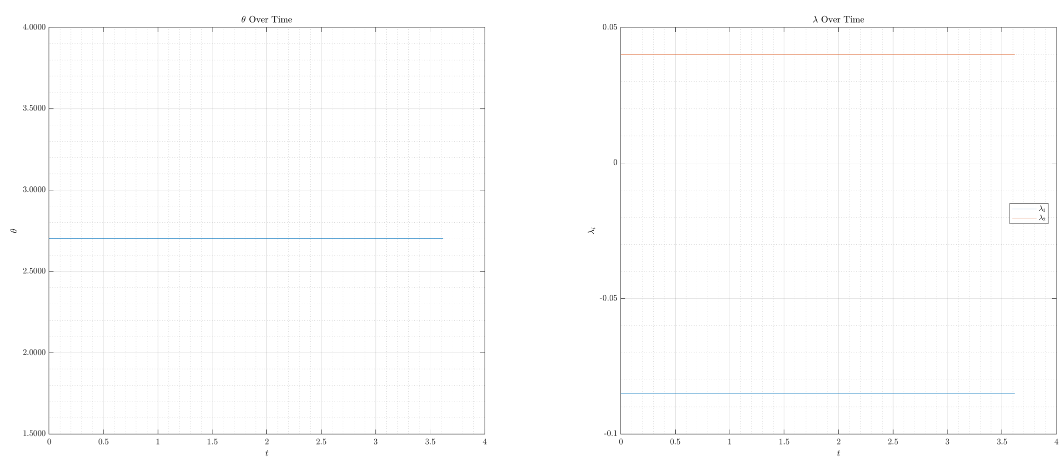

Figure 4: Problem 4(b) optimal control and costate

The red dotted line in Figure 4 is the shoreline and we can see that the final state is on it. Furthermore we can see that the control and the Hamiltonian are both constants as expected. The Hamiltonian is bouncing at a very small range that can be approximated to a constant of 0. The code used to accomplish this problem is in ??.

(c) For this problem the dynamics constraint is updated to

$$\dot{x} = V\cos\theta - (y - 50)$$
$$\dot{y} = V\sin\theta + 2(x - 15)$$

and the costates are

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x} = -2\lambda_2$$
$$\dot{\lambda}_2 = -\frac{\partial H}{\partial x} = \lambda_1$$

the optimal control remains the same. The transversality conditions and the terminal constraint are the same as problem (b). We can solve this numerically using `MATLAB`. The numerical approach taken is identical to problem 4(b) and refer to the code in ??.

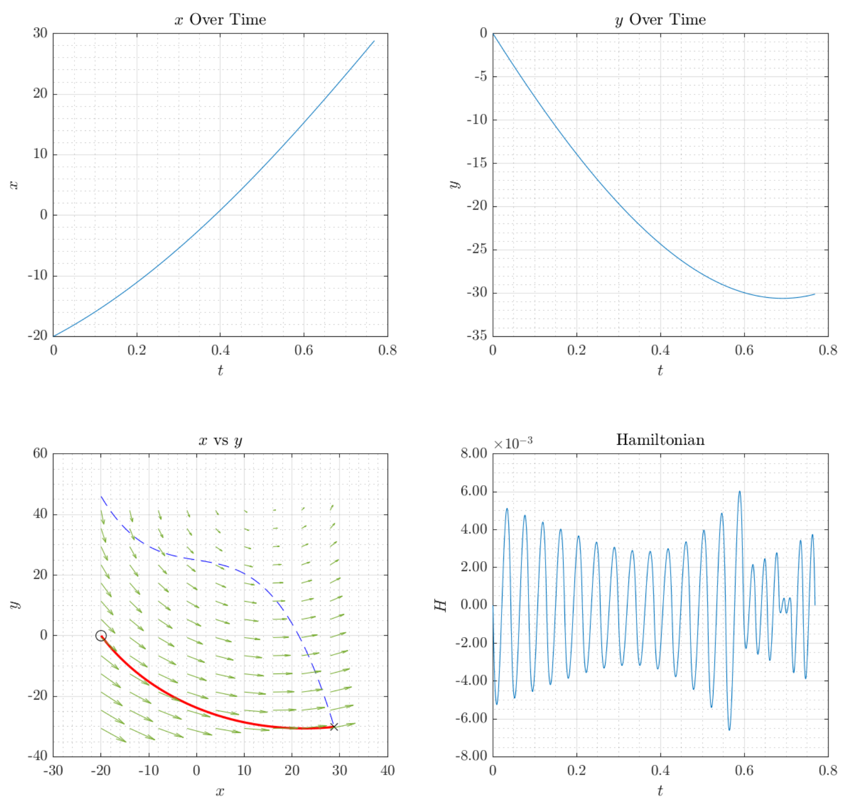

Figure 5: Problem 4(c) optimal trajectories and the Hamiltonian

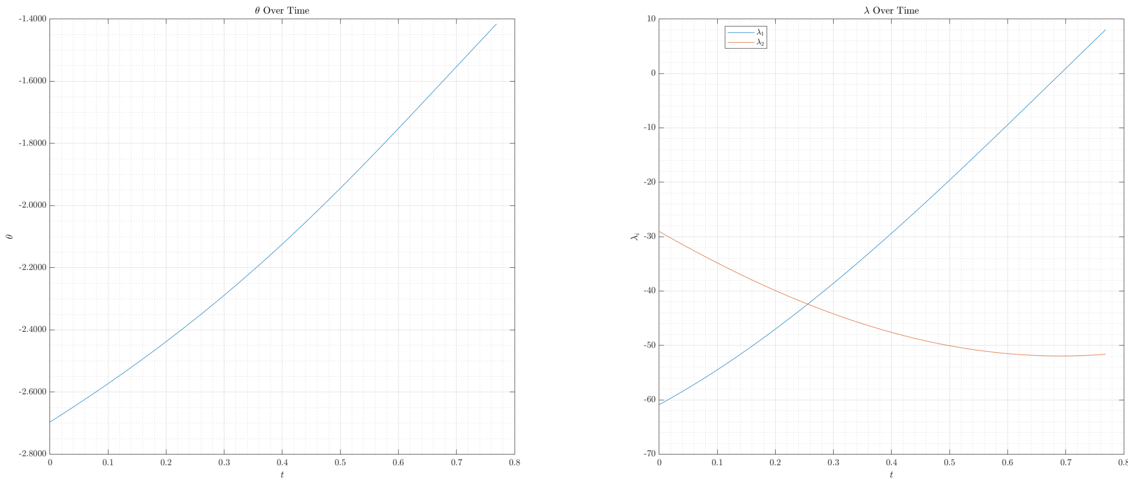

Figure 6: Problem 4(c) optimal control

In Figure 5 we can see that the final state lies on the shoreline and the Hamiltonian fluctuates at a extremely small range which means that it is very close to being a constant 0. The resulting minimal time is

$$\boxed{\min t_f = 0.7687.}$$

Problem 3

Investigate the existence of conjugate points in the closed interval $(0, t_f]$ for all values of $t_f > 0$, for the problem

$$\min J(u) = \int_0^{t_f} \left[t^2 + x^2(t) + u^2(t)\right] dt$$

where $\dot{x}(t) = u(t)$ and the boundary conditions are $x(0) = 0$ and $x(t_f) = 0$.

Solution:
We can rewrite this problem as

$$J = \frac{t_f^3}{3} + \int_0^{t_f} \left(x^2(t) + u^2(t)\right) dt$$

and the Hamiltonian becomes

$$H = x^2 + u^2 + \lambda u.$$

The optimal control for this problem becomes

$$\begin{aligned} \frac{\partial H}{\partial u} &= 2u + \lambda = 0 \\ u &= -\frac{\lambda}{2}. \end{aligned}$$

For this problem, note that we have $H_{uu} = 2 > 0$ and the strengthened form of the Legendre condition is satisfied which implies that the control is regular as well as the trajectory. If we investigate the accessory minimization problem

$$\begin{aligned} J_{acc}(v) = \frac{1}{2} \int_0^T &\left[\delta x^T(t) R_1(t) \delta x(t) + 2\delta x^T(t) R_{12}(t) v(t) + v^T(t) R_2(t) v(t)\right] dt \\ \dot{x} &= A(t)x(t) + B(t)u(t) \end{aligned}$$

we know that $R_1 = -\alpha$, $R_{12} = 0$, $R_2 = 1$, $A = 1$, and $B = 1$. Then,

$$\begin{aligned} \tilde{A} &= A - BR_2^{-1}R_{12} = 0 \\ \Sigma &= BR_2^{-1}B^T = \frac{1}{2} \\ \tilde{R} &= R_1 - R_{12}R_2^{-1}R_{12}^T = 2 \end{aligned}$$

Hence, the Riccati equation for this problem is

$$\dot{s}(t) = -\frac{1}{2}s^2(t) + 2.$$

This can be solved to be

$$s(t) = -2\tanh(c - t).$$

Applying the boundary condition of $s(t_f) = 0$

$$s(t) = -2\tanh(t_f - t).$$

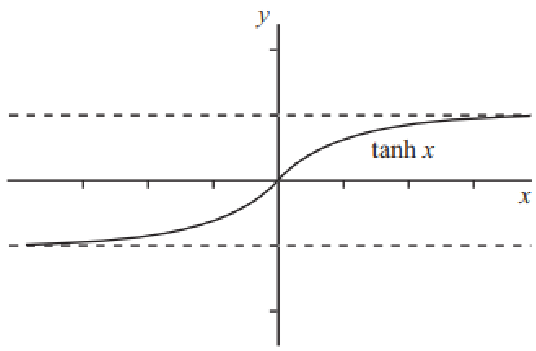

Figure 7: tanh function

From this we can see that there exists no conjugate point for this problem.

Problem 4

Show that the matrix Riccati differential equation

$$-\dot{P}(t) = A^T P(t) + P(t)A - P(t)BR_2^{-1}B^T P(t)$$

with initial condition

$$P(0) = P_0$$

where P_0 is nonsingular, has the solution

$$P(t) = e^{-A^T t}\left(P_0^{-1} - \int_0^t e^{-As}BR_2^{-1}B^T e^{-A^T s}ds\right)^{-1} e^{-At}.$$

Solution:
Assuming the matrices $P(t)$ and R_2 are symmetric positive definite, we multiply the given matrix Riccati differential equation by $P^{-1}(t)$ from both sides which gives

$$-P^{-1}(t)\dot{P}(t)P^{-1}(t) = P^{-1}(t)A^T + AP^{-1}(t) - BR_2^{-1}B^T.$$

Also we know that

$$\frac{d}{dt}P^{-1}(t) = -P^{-1}(t)\frac{dP(t)}{dt}P^{-1}(t)$$

then the equation above becomes

$$\frac{d}{dt}P^{-1}(t) = P^{-1}(t)A^T + AP^{-1}(t) - BR_2^{-1}B^T.$$

For the next procedures we will use the concepts and properties of vectorization, kronecker product, and direct sum. First, letting $Z(t) = P^{-1}(t)$ and $V = BR_2^{-1}B^T$ we can vectorize the above equation. But this is possible since we have assumed R_2 to be symmetric positive definite and $R_2 = MM^T$ if M is a real nonsingular matrix, which results in the fact that

$$V = BMM^T B^T = \left\|M^T B^T\right\|^2 \geq 0.$$

Thus, if we use vectorization we have

$$\begin{aligned}\text{vec}\left(\dot{Z}(t)\right) &= \text{vec}\left(Z(t)A^T + AZ(t)\right) - \text{vec}\,(V) \\ &= \text{vec}\left(IZ(t)A^T + AZ(t)I\right) - \text{vec}\,(V)\end{aligned}$$

and if we use the property of

$$\text{vec}\,(ABC) = \left(C^T \otimes A\right)\text{vec}\,(B)$$

then

$$\operatorname{vec}\left(\dot{Z}(t)\right) = (A \otimes I)\operatorname{vec}(Z(t)) + \left(I^T \otimes A\right)\operatorname{vec}(Z(t)) - \operatorname{vec}(V).$$

Now if we apply the property of Kronecker product

$$A \otimes I + I \otimes B = A \oplus B$$

we have

$$\operatorname{vec}\left(\dot{Z}(t)\right) = (A \oplus A)\operatorname{vec}(Z(t)) - \operatorname{vec}(V).$$

This shape resembles the common shape of the differential equation of $\dot{x}(t) = Ax + b$ which can be solved in the form of

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-s)}b(s)ds.$$

Similarly, we can solve the above equation as

$$\begin{aligned}
\operatorname{vec}(Z(t)) &= e^{(A\oplus A)t}\operatorname{vec}(Z(0)) - \int_0^t e^{(A\oplus A)(t-s)}\operatorname{vec}(V)ds \\
&= e^{(A\oplus A)t}\operatorname{vec}(Z(0)) - e^{(A\oplus A)t}\int_0^t e^{-(A\oplus A)s}\operatorname{vec}(V)ds \\
&= e^{(A\oplus A)t}\left[\operatorname{vec}(Z(0)) - \int_0^t e^{-(A\oplus A)s}\operatorname{vec}(V)ds\right]
\end{aligned}$$

if we apply the property of

$$e^{A\oplus B} = e^A \otimes e^B$$

we have

$$\operatorname{vec}(Z(t)) = e^{At} \otimes e^{At}\left[\operatorname{vec}(Z(0)) - \int_0^t e^{-As} \otimes e^{-As}\operatorname{vec}(V)ds\right]$$

now if we reverse the property

$$\left(C^T \otimes A\right)\operatorname{vec}(B) = \operatorname{vec}(ABC)$$

we can rewrite the equation into

$$\begin{aligned}
\operatorname{vec}(Z(t)) &= e^{At} \otimes e^{At}\left[\operatorname{vec}(Z(0)) - \operatorname{vec}\left(\int_0^t e^{-As}Se^{-A^Ts}ds\right)\right] \\
&= e^{At} \otimes e^{At}\operatorname{vec}\left(Z(0) - \int_0^t e^{-As}Se^{-A^Ts}ds\right) \\
&= \operatorname{vec}\left(e^{At}\left[Z(0) - \int_0^t e^{-As}Se^{-A^Ts}ds\right]e^{A^Tt}\right).
\end{aligned}$$

Then, we remove the vectorization from both sides and we have

$$Z(t) = e^{At}\left[Z(0) - \int_0^t e^{-As} S e^{-A^T s} ds\right] e^{A^T t}$$

and since $P(t) = Z^{-1}(t)$ and $P^{-1}(0) = Z(0)$ we have

$$P(t) = \left(e^{At}\left[P^{-1}(0) - \int_0^t e^{-As} S e^{-A^T s} ds\right] e^{A^T t}\right)^{-1}$$

and hence

$$\boxed{P(t) = e^{-A^T t}\left(P_0^{-1} - \int_0^t e^{-As} B R_2^{-1} B^T e^{-A^T s} ds\right)^{-1} e^{-At}.}$$

$\blacksquare$

Problem 5

Consider the harmonic oscillator

$$\begin{aligned}\dot{x}_1(t) &= x_2(t)\\ \dot{x}_2(t) &= -x_1(t) + u(t)\end{aligned}$$

with the energy performance measure

$$J(u) = \int_0^{t_f} u^2(t)dt$$

Let $x_1(0) = 1$ and $x_2(0) = 2$, and require that $x_1(t_f) = 0$ and $x_2(t_f) = 0$. Consider three cases $t_f = 10, 5, 1$. For each case, simulate the controlled system with the minimum energy controller and compute the control effort as measured by

$$\left(\int_0^{t_f} u^2(t)dt\right)^{\frac{1}{2}}, \qquad \int_0^{t_f} |u(t)|dt, \qquad \max_{t\in[0,t_f]} |u(t)|$$

Solution:
The Hamiltonian of this problem is

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

The costate equations are

$$\begin{aligned}\dot{\lambda}_1 &= \lambda_2\\ \dot{\lambda}_2 &= -\lambda_1\end{aligned}$$

and the optimal control becomes

$$\begin{aligned}\frac{\partial H}{\partial u} &= 2u + \lambda_2 = 0\\ u &= -\frac{\lambda_2}{2}.\end{aligned}$$

Now if we solve the costates as a systems ordinary differential equation we have the following

$$\begin{aligned}\lambda_1 &= -c_1 \sin t + c_2 \cos t\\ \lambda_2 &= c_1 \cos t + c_2 \sin t\end{aligned}$$

then

$$u = -\frac{c_1}{2}\cos t - \frac{c_2}{2}\sin t.$$

And we have

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - \frac{c_1}{2}\cos t - \frac{c_2}{2}\sin t\end{aligned}$$

this nonhomogeneous system ODE is solvable and the result is the following

$$\begin{aligned}x_1(t) &= c_3\, \cos(t) - 0.2500\, c_2\, \cos(t) + c_4\, \sin(t) - 0.2500\, c_1\, t\, \cos(t) - 0.2500\, c_2\, t\, \sin(t) \\ x_2(t) &= c_4\, \cos(t) - 0.2500\, c_1\, \cos(t) - c_3\, \sin(t) - 0.2500\, c_2\, t\, \cos(t) + 0.2500\, c_1\, t\, \sin(t)\end{aligned}$$

Now we are able to apply the boundary conditions and the t_f value to find the 4 unknowns. The resulting plots and the control efforts are shown below.

$\underline{t_f = 10}$

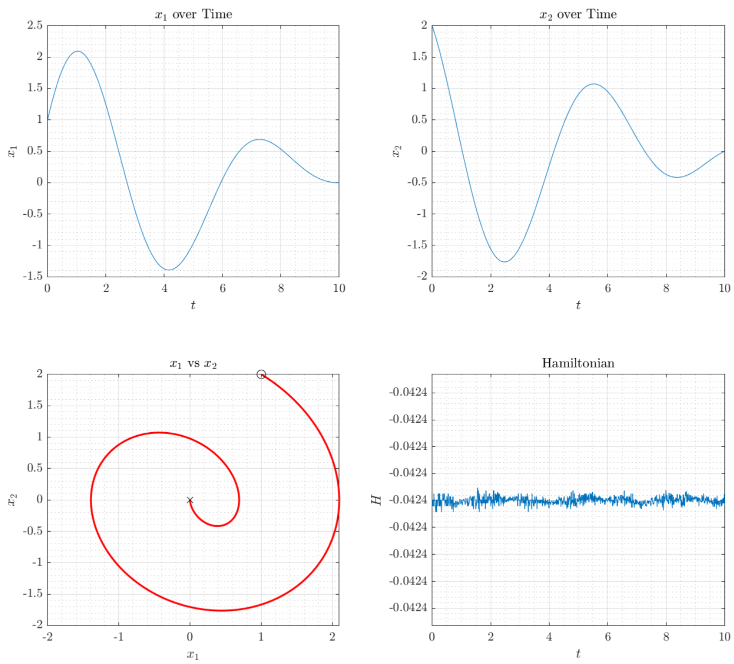

Figure 8: Optimal path, trajectory, and Hamiltonian when $t_f = 10$

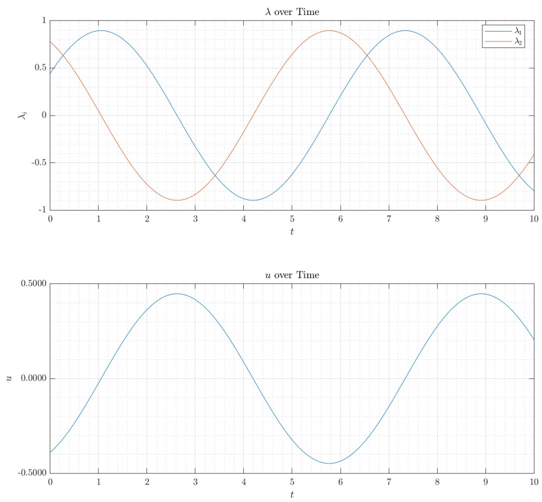

Figure 9: Optimal control and costates when $t_f = 10$

$t_f = 5$

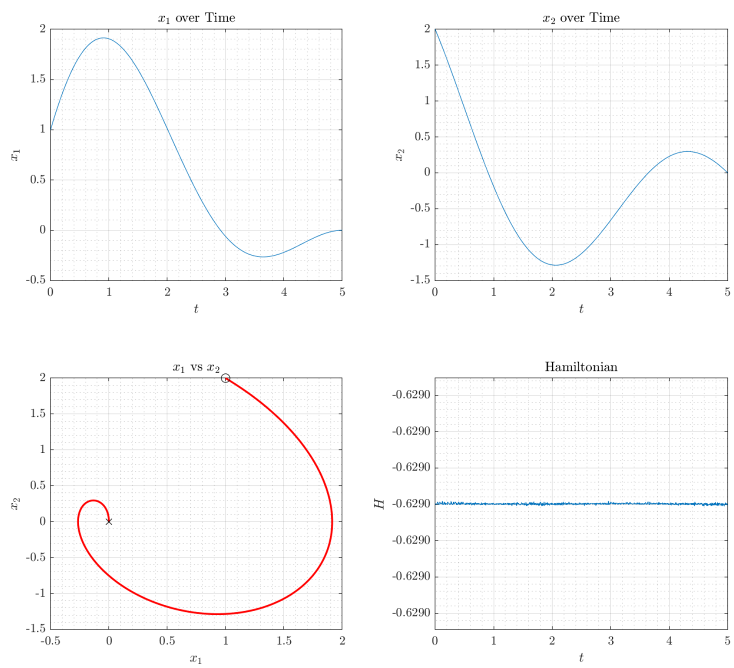

Figure 10: Optimal path, trajectory, and Hamiltonian when $t_f = 5$

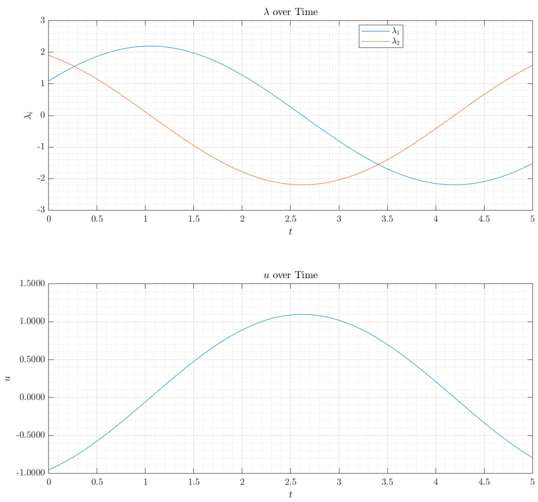

Figure 11: Optimal control and costates when $t_f = 5$

$\underline{t_f = 1}$

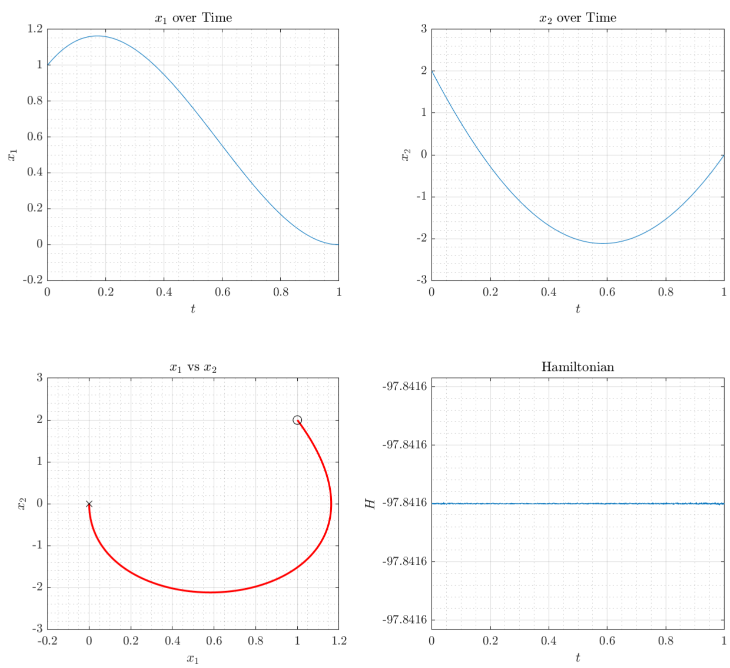

Figure 12: Optimal path, trajectory, and Hamiltonian when $t_f = 1$

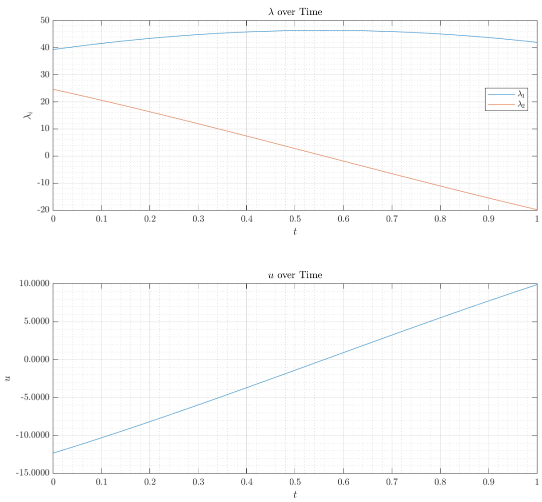

Figure 13: Optimal control and costates when $t_f = 1$

t_f	10	5	1
$\left(\int_0^{t_f} u^2(t)dt\right)^{\frac{1}{2}}$	0.9996	1.5651	6.6570
$\int_0^{t_f} \lvert u(t)\rvert dt$	2.8610	3.0854	5.7552
$\max_{t\in[0,t_f]} \lvert u(t)\rvert$	0.4475	1.0973	12.3235

Table 1: Control effort table for each t_f

Problem 6

Consider the problem of a spacecraft attempting to make a soft landing on the moon using the minimum amount of fuel. The equations of motion are

$$\begin{aligned} \dot{h} &= v \\ \dot{v} &= -g + \frac{u}{m} \\ \dot{m} &= -cu \end{aligned}$$

where m denotes the mass, h the alititude, v the velocity, and u is the thrust of the spacecraft's engine, c a constant, and g the gravity acceleration of the moon (considered to be constant). The control u is restricted so that $0 \leq u(t) \leq u_{max} = 1$.

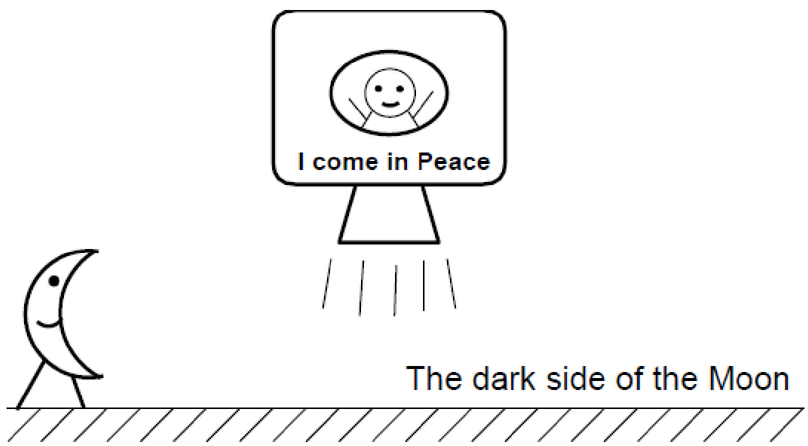

Figure 14: Soft moon landing diagram with the iconic banana man

The boundary conditions are

$$\begin{gathered} h(0) = h_0, \qquad v(0) = v_0, \qquad m(0) - m_{net} - m_0 = 0 \\ h(T) = v(T) = 0 \end{gathered}$$

where m_{net} denotes the mass of the spacecraft without fuel, h_0 the initial height, v_0 the initial velocity, m_0 the initial amount of fuel, and T is the (given) time taken for touchdown. Solve this optimal control problem, that is, minimize

$$J(u) = \int_0^T u(t)dt$$

subject to the previous dynamic and boundary constraints.

Solution:
The Hamiltonian for this problem becomes

$$H = u + \lambda_1(v) + \lambda_2(-g + \frac{u}{m}) + \lambda_3(-cu).$$

and the costates are

$$\begin{aligned}
\dot{\lambda}_1 &= 0 \\
\dot{\lambda}_2 &= -\lambda_1 \\
\dot{\lambda}_3 &= \frac{\lambda_2 u}{m^2}
\end{aligned}$$

and the optimal control for this problem

$$\frac{\partial H}{\partial u} = 1 + \frac{\lambda_2}{m} - \lambda_3 c = 0.$$

Observing the Hamiltonian we can see that the Hamiltonian is linearly proportional to u and thus we can apply Pontraygin's Maximum Principle to solve this optimal control problem. Therefore,

$$\begin{aligned}
u &= 0 \qquad \text{if} \qquad 1 + \frac{\lambda_2}{m} - \lambda_3 c > 0 \\
u &= 1 \qquad \text{if} \qquad 1 + \frac{\lambda_2}{m} - \lambda_3 c < 0
\end{aligned}$$

and for this problem we apply the control of at the initial conditions we have the control to be $u = 0$ and at the final conditions we have the control to be $u = 1$ which is a bang-bang control application. However, before we implement this, we must check the singular control conditions.

$$\begin{aligned}
\dot{H}_u &= \frac{\dot{\lambda}_2 m - \lambda_2 \dot{m}}{m^2} - \dot{\lambda}_3 c \\
&= -\frac{\lambda_1}{m} - \frac{\lambda_2}{m^2}(-cu) - \frac{\lambda_2 uc}{m^2} \\
&= \frac{\lambda_1}{m}
\end{aligned}$$

$$\begin{aligned}
\ddot{H}_u &= -\frac{\dot{\lambda}_1 m - \lambda_1 \dot{m}}{m^2} \\
&= -\frac{\lambda_1 c}{m^2} u
\end{aligned}$$

Then, the GLC condition shows

$$(-1)\frac{\partial}{\partial u}\left[\frac{d^2}{dt^2}\left(\frac{\partial H}{\partial u}\right)\right] = \frac{\lambda_1 c}{m^2}$$

and this can be treated negative if we assume $\lambda_1 = const. < 0$, which implies that there is no optimal singular control and we can ignore singular conditions.

Now if $u = 0$

$$\begin{aligned}
\dot{h} &= v \\
\dot{v} &= -g \\
\dot{m} &= 0 \\
\dot{\lambda}_1 &= 0 \\
\dot{\lambda}_2 &= -\lambda_1 \\
\dot{\lambda}_3 &= 0
\end{aligned}$$

If we solve these differential equations we have the following

$$\begin{aligned}
\lambda_1 &= c_1 \\
\lambda_2 &= -c_1 t + c_2 \\
m &= c_3 \\
\lambda_3 &= c_4 \\
v &= -gt + c_5 \\
h &= -\frac{1}{2} g t^2 + c_5 t + c_6 \\
H &= c_1(-gt + c_5) - (-c_1 t + c_2) g
\end{aligned}$$

now if we apply the initial conditions to the equations we have

$$\begin{aligned}
c_3 &= m_{net} + m_0 = \tilde{m} \\
c_5 &= v_0 \\
c_6 &= h_0
\end{aligned}$$

Then the optimal trajectories becomes

$$h = -\frac{g}{2} t^2 + v_0 t + h_0 \tag{1}$$

$$v = -gt + v_0 \tag{2}$$

$$m = \tilde{m} = m_{net} + m_0 \tag{3}$$

Now for the second section (past the switching point up to the final condition) we assume that the control has a value of $u = 1$ which gives the following equations

$$\begin{aligned}
\lambda_1 &= c_1 \\
\lambda_2 &= -c_1 t + c_2 \\
\lambda_3 &= c_4 + \frac{c\, c_2 - c_1\, c_3}{c^2\,(c_3 - c\, t)} - \frac{c_1\, \ln\left(c\, t - c_3\right)}{c^2}
\end{aligned}$$

and

$$m = -ct + c_3 \tag{4}$$

$$v = c_5 - g\,t - \frac{\ln\left(c_3 - c\,t\right)}{c} \tag{5}$$

$$h = c_6 + c_5\,t + \frac{t}{c} - \frac{g\,t^2}{2} + \frac{c_3\,\ln\left(c_3 - c\,t\right)}{c^2} - \frac{t\,\ln\left(c_3 - c\,t\right)}{c} \tag{6}$$

Now if we apply the final conditions to this equations and let $m(T) = m_f$ we have

$$\begin{aligned} m(0) &= m_f = -cT + c_3 \\ v(0) &= 0 = -gT - \frac{1}{c}\ln(c_3 - cT) + c_5 \\ h(0) &= 0 = -\frac{g}{2}T^2 + c_5T + \frac{1}{c}\left[T - T\ln(c_3 - cT)\right] + \frac{c_3}{c^2}\ln(c_3 - cT) + c_6 \end{aligned}$$

and therefore

$$\begin{aligned} c_3 &= m_f + cT \\ c_5 &= gT + \frac{1}{c}\ln(c_3 - cT) \\ c_6 &= \frac{g}{2}T^2 - c_5T - \frac{1}{c}\left[T - T\ln(c_3 - cT)\right] - \frac{c_3}{c^2}\ln(c_3 - cT). \end{aligned}$$

Now at a certain switching time t_s the equations for the first section and the second section are equal, and thus, we can equate the equations (3)=(4) and (2)=(5) (the equation for altitude is unnecessary since it is a direct integration of the velocity state and if the velocities are equal the altitude will also be equal) at $t = t_s$ with the unknowns being m_f and t_s. Thus, we obtain

$$\boxed{\begin{aligned} t_s &= -\frac{m_0 + m_{\mathrm{net}} - \mathrm{e}^{c\left(v_0 - T\,g + \frac{\log\left(m_0 + m_{\mathrm{net}}\right)}{c}\right)} - T\,c}{c} \\ m_f &= \mathrm{e}^{c\left(v_0 - T\,g + \frac{\log\left(m_0 + m_{\mathrm{net}}\right)}{c}\right)} \end{aligned}}$$

Problem 7

Consider the following problem

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

subject to

$$\dot{x} = x + u, \qquad x(0) = x(T) = 0$$

(a) Find the extremal control and the extremal trajectory for this problem for $\alpha > 1$.

(b) Investigate the existence (and location) of conjugate points for $\alpha = 2$ and $T = \pi$. **Hint:** Investigate the accessory minimization problem.

Solution:
(a) The Hamiltonian of this problem is

$$H = \frac{1}{2}u^2 - \frac{1}{2}\alpha x^2 + \lambda(x+u).$$

The costate equation is

$$\dot{\lambda} = \alpha x - \lambda$$

and the optimal control is

$$\frac{\partial H}{\partial u} = u + \lambda = 0$$
$$u = -\lambda.$$

Now we can solve the following system ODE

$$\begin{bmatrix} \dot{\lambda} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -1 & \alpha \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda \\ x \end{bmatrix}.$$

Thus, we have

$$\begin{aligned} x(t) &= c_2 \sin\left(t\sqrt{\alpha-1}\right) + c_1 \cos\left(t\sqrt{\alpha-1}\right) \\ \lambda(t) &= c_1\left(\cos\left(t\sqrt{\alpha-1}\right) + \sin\left(t\sqrt{\alpha-1}\right)\sqrt{\alpha-1}\right) \\ &\quad + c_2\left(\sin\left(t\sqrt{\alpha-1}\right) - \cos\left(t\sqrt{\alpha-1}\right)\sqrt{\alpha-1}\right) \end{aligned}$$

which gives us

$$\begin{aligned} u(t) &= -c_1\left(\cos\left(t\sqrt{\alpha-1}\right) + \sin\left(t\sqrt{\alpha-1}\right)\sqrt{\alpha-1}\right) \\ &\quad - c_2\left(\sin\left(t\sqrt{\alpha-1}\right) - \cos\left(t\sqrt{\alpha-1}\right)\sqrt{\alpha-1}\right). \end{aligned}$$

by applying the boundary condition of $x(0) = 0$ we know that $c_1 = 0$, and therefore

$$\begin{aligned}
x(t) &= c_2 \sin\left(t\sqrt{\alpha - 1}\right) \\
\lambda(t) &= c_2 \left(\sin\left(t\sqrt{\alpha - 1}\right) - \sqrt{\alpha - 1}\cos\left(t\sqrt{\alpha - 1}\right)\right) \\
u(t) &= -c_2 \left(\sin\left(t\sqrt{\alpha - 1}\right) - \sqrt{\alpha - 1}\cos\left(t\sqrt{\alpha - 1}\right)\right)
\end{aligned}$$

At the final time T, x is supposed to be 0 based on the boundary condition which is only possible when

$$c_2 = 0 \qquad or \qquad T = \frac{n\pi}{\sqrt{\alpha - 1}}$$
$$\text{where } n = 0, 1, 2, \ldots$$

For the former condition x, u, and H are all zero and for the latter $c_2 \in \mathbb{R}$ so we let $c_2 = 1$. Hence, the optimal trajectory and the optimal control are

$$\boxed{\begin{aligned}
x(t) &= \sin\left(t\sqrt{\alpha - 1}\right) \\
u(t) &= -\left(\sin\left(t\sqrt{\alpha - 1}\right) - \sqrt{\alpha - 1}\cos\left(t\sqrt{\alpha - 1}\right)\right)
\end{aligned}}$$

(b) For this problem, note that we have $H_{uu} = 1 > 0$ and the strengthened form of the Legendre condition is satisfied which implies that the control is regular as well as the trajectory. If we investigate the accessory minimization problem

$$J_{acc}(v) = \frac{1}{2}\int_0^T \left[\delta x^T(t)R_1(t)\delta x(t) + 2\delta x^T(t)R_{12}(t)v(t) + v^T(t)R_2(t)v(t)\right]dt$$
$$\dot{x} = A(t)x(t) + B(t)u(t)$$

we know that $R_1 = -\alpha$, $R_{12} = 0$, $R_2 = 1$, $A = 1$, and $B = 1$. Hence, the Riccati equation for this problem is

$$\dot{s}(t) = s^2(t) - 2s(t) + \alpha.$$

This can be solved to be

$$s(t) = 1 + \sqrt{\alpha - 1}\tan\left((c + t)\sqrt{\alpha - 1}\right).$$

Applying the boundary condition of $s(\pi) = 0$ and $\alpha = 2$

$$s(t) = 1 + \tan\left(t - \frac{\pi}{4}\right).$$

From this we can see that there exists a conjugate point at $t = 3\pi/4$.

Problem 8

Use the Maximum Principle to solve the following optimal control problem

$$\min \int_0^{\frac{\pi}{6}} \left(\frac{u^2(t)}{\cos(t)} - u(t) \right) dt$$

subject to

$$\dot{x}(t) = -u(t)$$

with boundary conditions

$$x(0) = 0, \qquad x\left(\frac{\pi}{6}\right) = -\frac{1}{8}$$

and control constraint $u(t) \geq 0$.

You may use the fact that $\cos(t) > 0$ on $t \in [0, \frac{\pi}{6}]$.

Solution:
The Hamiltonian of this problem is

$$H = \frac{u^2}{\cos t} - u + \lambda(-u),$$

and the costate

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = 0,$$

and the optimal control for this problem is

$$\begin{aligned} \frac{\partial H}{\partial u} &= 2\frac{u}{\cos t} - 1 - \lambda = 0 \\ u &= \frac{1}{2}(\lambda + 1)\cos t. \end{aligned}$$

Now from the costate we know that $\lambda = \text{const.}$ and we can let $\frac{1}{2}(\lambda + 1) = c = \text{const.}$ Additionally, in the range of $t \in [0, \pi/6]$, $\cos t > 0$ holds true so we will fixate $\sec t = \gamma > 0$ for convenience, and we can reorganize the expression of the Hamiltonian as follows.

$$\begin{aligned} H &= \gamma u^2 - u - \lambda u \\ &= \gamma \left(u - \frac{\lambda + 1}{2\gamma} \right)^2 - \frac{(\lambda + 1)^2}{4\gamma}, \end{aligned}$$

which can be visualized as such

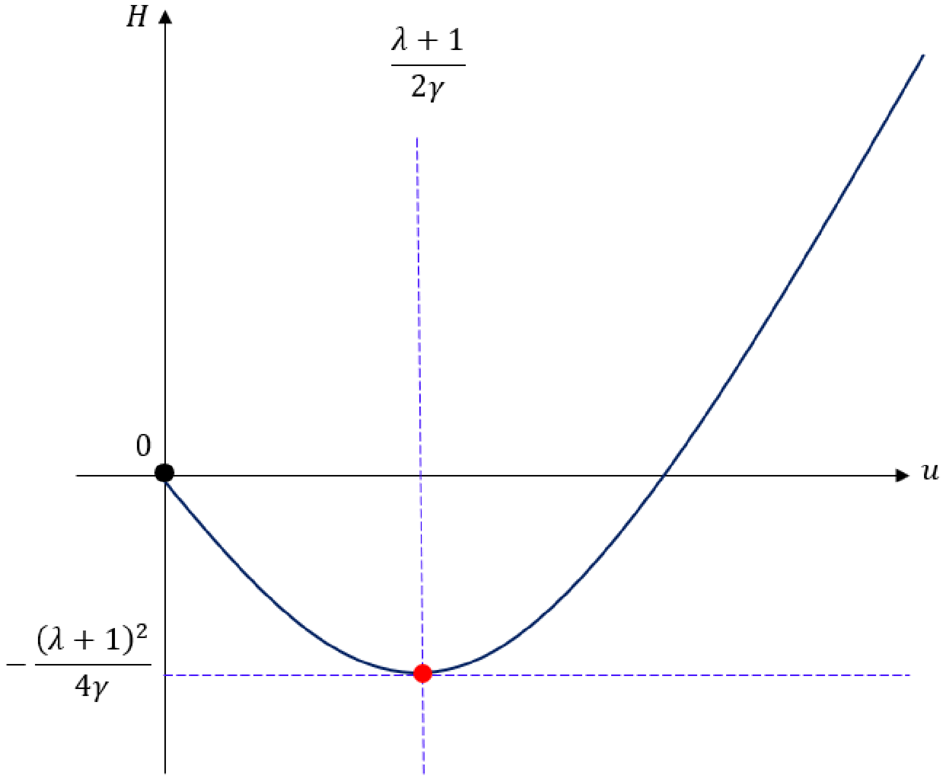

Figure 15: Hamiltonian over the control

From Pontryagin's Maximum Principle, we can tell from Figure 15 that the optimal control for this system is

$$u = \frac{\lambda+1}{2\gamma} = \frac{1}{2}(\lambda+1)\cos t$$

and therefore,

$$x(t) = -\frac{1}{2}(\lambda+1)\sin t + c_1$$

and if we apply the boundary conditions we have

$$c_1 = 0, \qquad \lambda = -\frac{1}{2}$$

and since $u \geq 0$, which means that

$$u = \frac{1}{2}(\lambda+1)\cos t \geq 0 \quad \longrightarrow \quad \lambda \geq -1$$

this condition is satisfied. Hence,

$$\boxed{\begin{aligned} u(t) &= \frac{1}{4}\cos t \\ x(t) &= -\frac{1}{4}\sin t \end{aligned}}$$

Appendix

9.1 Problem 2a: MATLAB Code

```
% AE6511 Hw5 Problem 4(a) MATLAB code
% Tomoki Koike
clear all; close all; clc;  % housekeeping commands
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%%
% BVP4C
V = 15;
x0 = [0, 0, 0.0055, -0.1121, 1];
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, V);

% Unpack results
T = linspace(0,1,1000);
[xopt,xdopt] = deval(sol,T);
xopt = xopt.'; xdopt = xdopt.';
tf_eval = mean(xopt(1,5));
tf_out = abs(tf_eval);
tspan = tf_out * T;
x_sol = xopt(:,1);
y_sol = xopt(:,2);
lambda1_sol = xopt(:,3);
lambda2_sol = xopt(:,4);
theta_sol = atan2(lambda2_sol,lambda1_sol);
H = (-1 + lambda1_sol .* xdopt(:,1)./tf_eval ...
    + lambda2_sol .* xdopt(:,2)/tf_eval);

% Phase portrait/Current
X_min = min(x_sol);
X_max = max(x_sol);
Y_min = min(y_sol);
Y_max = max(y_sol);
[X,Y] = meshgrid(X_min:1.6:X_max,Y_min:1.6:Y_max);
U = 2*ones(size(X));
V = -6*ones(size(Y));

%%
```

```
fig = figure("Renderer","painters","Position",[60 60 900 800]);
    % x vs t
    subplot(2,2,1)
    plot(tspan, x_sol)
    title('$x$ Over Time')
    xlabel('$t$')
    ylabel('$x$')
    grid on; grid minor; box on;
    % y vs t
    subplot(2,2,2)
    plot(tspan, y_sol)
    title('$y$ Over Time')
    xlabel('$t$')
    ylabel('$y$')
    grid on; grid minor; box on;
    % x - y
    subplot(2,2,3)
    plot(x_sol, y_sol)
    hold on;
    quiver(X,Y,U,V)
    title('$x$ vs $y$')
    xlabel('$x$')
    ylabel('$y$')
    grid on; grid minor; box on; hold off;
    % Hamiltonian
    subplot(2,2,4)
    plot(tspan, H)
    title('Hamiltonian')
    xlabel('$t$')
    ylabel('$H$')
    ytickformat('%,.2f')
    grid on; grid minor; box on;
saveas(fig, 'p4a.png');
%%
% Plot costates
fig = figure("Renderer","painters","Position",[60 60 900 800]);
    plot(tspan, lambda1_sol,'DisplayName','$\lambda_1$')
    hold on;
    plot(tspan, lambda2_sol,'DisplayName','$\lambda_2$')
    title('$\lambda$ Over Time')
    xlabel('$t$')
    ylabel('$\lambda_i$')
    legend('Location','best'); grid on; grid minor; box on; hold off;
saveas(fig,'p4a_lambda.png');
%%
```

```
% Plot control
fig = figure("Renderer","painters","Position",[60 60 900 800]);
    plot(tspan, theta_sol)
    title('$\theta$ Over Time')
    xlabel('$t$')
    ylabel('$\theta$')
    ytickformat('%,.4f')
    grid on; grid minor; box on;
saveas(fig, 'p4a_theta.png');
%% Function

function dxdt = odefcn(t,x,V)
    dxdt = zeros(5,1);
    theta = atan2(x(4),x(3));
    dxdt(1) = V * cos(theta) + 2; % x
    dxdt(2) = V * sin(theta) - 6; % y
    dxdt(3) = 0; % lambda1
    dxdt(4) = 0; % lambda2
    dxdt(5) = 0; % tf
    dxdt = dxdt * x(5);
end

function res = bcfcn(xa,xb,V)
    res = zeros(5,1);
    theta_f = atan2(xb(4),xb(3));
    res(1) = xa(1) + 20; % x(0)
    res(2) = xa(2);  % y(0)
    res(3) = xb(1) + 15; % x(tf)
    res(4) = xb(2) - 35.5; % y(tf)
    res(5) = (-1 + xb(3) * (V*cos(theta_f)+2) ...
        + xb(4)*(V*sin(theta_f)-6))*xb(5); % H(t_f)
end
```

9.2 Problem 2b: MATLAB Code

```
% AE6511 Hw5 Problem 4(b) MATLAB code
% Tomoki Koike
clear all; close all; clc;  % housekeeping commands
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%%
% BVP4C
```

```matlab
V = 15;
x0 = [0, 0, 1, 1, 1];
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, V);

% Unpack results
T = linspace(0,1,1000);
[xopt,xdopt] = deval(sol,T);
xopt = xopt.'; xdopt = xdopt.';
tf_eval = mean(xopt(1,5));
tf_out = abs(tf_eval);
tspan = tf_out * T;
x_sol = xopt(:,1);
y_sol = xopt(:,2);
lambda1_sol = xopt(:,3);
lambda2_sol = xopt(:,4);
theta_sol = atan2(lambda2_sol,lambda1_sol);
H = (-1 + lambda1_sol .* xdopt(:,1)./tf_eval ...
    + lambda2_sol .* xdopt(:,2)/tf_eval);

% Shoreline
x_shore = linspace(min(x_sol),max(x_sol),1000);
y_shore = -0.002*x_shore.^3 - 0.25*x_shore + 25;

% Phase portrait/Current
X_min = min(min(x_sol),min(x_shore));
X_max = max(max(x_sol),max(x_shore));
Y_min = min(min(y_sol),min(y_shore));
Y_max = max(max(y_sol),max(y_shore));
[X,Y] = meshgrid(X_min:6:X_max,Y_min:6:Y_max);
U = 2*ones(size(X));
V = -6*ones(size(Y));
%%
fig = figure("Renderer","painters","Position",[60 60 900 800]);
    % x vs t
    subplot(2,2,1)
    plot(tspan, x_sol)
    title('$x$ Over Time')
    xlabel('$t$')
    ylabel('$x$')
    grid on; grid minor; box on;
    % y vs t
    subplot(2,2,2)
```

```
    plot(tspan, y_sol)
    title('$y$ Over Time')
    xlabel('$t$')
    ylabel('$y$')
    grid on; grid minor; box on;
    % x - y with current
    subplot(2,2,3)
    plot(x_sol, y_sol,'-r',LineWidth=1.5)
    hold on;
    plot(x_sol(1),y_sol(1),'ok',MarkerSize=7)
    plot(x_sol(end),y_sol(end),'xk',MarkerSize=7)
    plot(x_shore,y_shore, '--b')
    quiver(X,Y,U,V)
    title('$x$ vs $y$')
    xlabel('$x$')
    ylabel('$y$')
    grid on; grid minor; box on; hold off;
    % Hamiltonian
    subplot(2,2,4)
    plot(tspan, H)
    title('Hamiltonian')
    xlabel('$t$')
    ylabel('$H$')
    ytickformat('%,.2f')
    grid on; grid minor; box on;
saveas(fig, 'p4b.png');
%%
% Plot costates
fig = figure("Renderer","painters","Position",[60 60 900 800]);
    plot(tspan, lambda1_sol,'DisplayName','$\lambda_1$')
    hold on;
    plot(tspan, lambda2_sol,'DisplayName','$\lambda_2$')
    title('$\lambda$ Over Time')
    xlabel('$t$')
    ylabel('$\lambda_i$')
    legend('Location','best'); grid on; grid minor; box on; hold off;
saveas(fig,'p4b_lambda.png');
%%
% Plot control
fig = figure("Renderer","painters","Position",[60 60 900 800]);
    plot(tspan, theta_sol)
    title('$\theta$ Over Time')
    xlabel('$t$')
    ylabel('$\theta$')
    ytickformat('%,.4f')
```

```
    grid on; grid minor; box on;
saveas(fig, 'p4b_theta.png');
%% Function

function dxdt = odefcn(t,x,V)
    dxdt = zeros(5,1);
    theta = atan2(x(4),x(3));
    dxdt(1) = V * cos(theta) + 2; % x
    dxdt(2) = V * sin(theta) - 6; % y
    dxdt(3) = 0; % lambda1
    dxdt(4) = 0; % lambda2
    dxdt(5) = 0; % tf
    dxdt = dxdt * x(5);
end

function res = bcfcn(xa,xb,V)
    res = zeros(5,1);
    theta_f = atan2(xb(4),xb(3));
    res(1) = xa(1) + 20; % x(0)
    res(2) = xa(2);  % y(0)
    res(3) = ((-0.25 - 0.006 * xb(1)^2)*(xb(4)-2*(xb(2)-35.5) ...
        + xb(3) - 2*(xb(1)+15))); % -l = Psi_x
    res(4) = 25 - 0.25*xb(1) - 0.002*xb(1)^3 - xb(2); % Psi(t_f)
    res(5) = (-1 + xb(3) * (V*cos(theta_f)+2) ...
        + xb(4)*(V*sin(theta_f)-6)) * xb(5); % H(t_f)
end
```

9.3 Problem 2c: MATLAB Code

```
% AE6511 Hw5 Problem 4(c) MATLAB code
% Tomoki Koike
clear all; close all; clc;  % housekeeping commands
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%%
% BVP4C
V = 15;
x0 = [0, 0, 2, 3, 1];
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, V);
```

```matlab

% Unpack results
T = linspace(0,1,50000);
[xopt,xdopt] = deval(sol,T);
xopt = xopt.'; xdopt = xdopt.';
tf_eval = mean(xopt(1,5));
tf_out = abs(tf_eval);
tspan = tf_out * T;
x_sol = xopt(:,1);
y_sol = xopt(:,2);
lambda1_sol = xopt(:,3);
lambda2_sol = xopt(:,4);
theta_sol = atan2(lambda2_sol,lambda1_sol);
H = (-1 + lambda1_sol .* xdopt(:,1)./tf_eval ...
    + lambda2_sol .* xdopt(:,2)/tf_eval);

% Shoreline
x_shore = linspace(min(x_sol),max(x_sol),length(tspan));
y_shore = -0.002*x_shore.^3 - 0.25*x_shore + 25;

% Phase portrait/Current
X_min = min(min(x_sol),min(x_shore));
X_max = max(max(x_sol),max(x_shore));
Y_min = min(min(y_sol),min(y_shore));
Y_max = max(max(y_sol),max(y_shore));
[X,Y] = meshgrid(X_min:6:X_max,Y_min:6:Y_max);
U = -Y + 50;
V = 2*X - 30;
%%
fig = figure("Renderer","painters","Position",[60 60 900 800]);
    % x vs t
    subplot(2,2,1)
    plot(tspan, x_sol)
    title('$x$ Over Time')
    xlabel('$t$')
    ylabel('$x$')
    grid on; grid minor; box on;
    % y vs t
    subplot(2,2,2)
    plot(tspan, y_sol)
    title('$y$ Over Time')
    xlabel('$t$')
    ylabel('$y$')
    grid on; grid minor; box on;
    % x - y
```

```matlab
    subplot(2,2,3)
    plot(x_sol, y_sol,'-r',LineWidth=1.5)
    hold on;
    plot(x_sol(1),y_sol(1),'ok',MarkerSize=7)
    plot(x_sol(end),y_sol(end),'xk',MarkerSize=7)
    plot(x_shore,y_shore, '--b')
    quiver(X,Y,U,V)
    title('$x$ vs $y$')
    xlabel('$x$')
    ylabel('$y$')
    grid on; grid minor; box on; hold off;
    % Hamiltonian
    subplot(2,2,4)
    plot(tspan, H)
    title('Hamiltonian')
    xlabel('$t$')
    ylabel('$H$')
    ytickformat('%,.2f')
    grid on; grid minor; box on;
saveas(fig, 'p4c.png');
%%
% Plot costates
fig = figure("Renderer","painters","Position",[60 60 900 800]);
    plot(tspan, lambda1_sol,'DisplayName','$\lambda_1$')
    hold on;
    plot(tspan, lambda2_sol,'DisplayName','$\lambda_2$')
    title('$\lambda$ Over Time')
    xlabel('$t$')
    ylabel('$\lambda_i$')
    legend('Location','best'); grid on; grid minor; box on; hold off;
saveas(fig,'p4c_lambda.png');
%%
% Plot control
fig = figure("Renderer","painters","Position",[60 60 900 800]);
    plot(tspan, theta_sol)
    title('$\theta$ Over Time')
    xlabel('$t$')
    ylabel('$\theta$')
    ytickformat('%,.4f')
    grid on; grid minor; box on;
saveas(fig, 'p4c_theta.png');
%% Function

function dxdt = odefcn(t,x,V)
    dxdt = zeros(5,1);
```

```
    theta = atan2(x(4),x(3));
    dxdt(1) = V * cos(theta) - x(2) + 50; % x
    dxdt(2) = V * sin(theta) + 2*x(1) - 30; % y
    dxdt(3) = -2*x(4); % lambda1
    dxdt(4) = x(3); % lambda2
    dxdt(5) = 0; % tf
    dxdt = dxdt * x(5);
end

function res = bcfcn(xa,xb,V)
    res = zeros(5,1);
    theta_f = atan2(xb(4),xb(3));
    res(1) = xa(1) + 20; % x(0)
    res(2) = xa(2);  % y(0)
    res(3) = ((-0.25 - 0.006 * xb(1)^2)*(xb(4)-2*(xb(2)-35.5) ...
        + xb(3) - 2*(xb(1)+15))); % -l = Psi_x
    res(4) = (25 - 0.25*xb(1) - 0.002*xb(1)^3 - xb(2)); % Psi(t_f)
    res(5) = (-1 + xb(3) * (V*cos(theta_f)-xb(2)+50) ...
        + xb(4)*(V*sin(theta_f)+2*xb(1)-30)) * xb(5); % H(t_f)
end
```

9.4 Problem 5: MATLAB Code

```
% AE6511 Hw6 Problem 5 MATLAB code
% Tomoki Koike
clear all; close all; clc;  % housekeeping commands
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%%
% Solving lambda
syms lambda_1(t) lambda_2(t)
X = [lambda_1; lambda_2];
A = [0 1; -1 0];
[l1sol(t),l2sol(t)] = dsolve(diff(X) == A*X);
%%
l1sol(t)
l2sol(t)
%%
% Solving x
syms x_1(t) x_2(t)
u = -l2sol(t)/2;
X = [x_1; x_2];
```

```matlab
M = [0 1; -1 0];
g = [0; u];
[x1sol(t),x2sol(t)] = dsolve(diff(X) == M*X + g);
%%
simplify(x1sol(t))
simplify(x2sol(t))
clear;
%%
% Then solve for the coefficients
% tf = 10
res1 = coeff_solver(10);
temp = struct2cell(res1);
cs1 = double([temp{:}]).';
CE1 = plotter(10,cs1,"outputs/p5_tf=10.png", ...
    "outputs/p5_control_tf=10.png");
%%
% tf = 5
res2 = coeff_solver(5);
temp = struct2cell(res2);
cs2 = double([temp{:}]).';
CE2 = plotter(5,cs2,"outputs/p5_tf=5.png", ...
    "outputs/p5_control_tf=5.png");
%%
% tf = 1
res3 = coeff_solver(1);
temp = struct2cell(res3);
cs3 = double([temp{:}]).';
CE3 = plotter(1,cs3,"outputs/p5_tf=1.png", ...
    "outputs/p5_control_tf=1.png");
%%
function res = coeff_solver(t_f)
    syms t c_1 c_2 c_3 c_4
    assume(c_1, 'real');
    assume(c_2, 'real');
    assume(c_3, 'real');
    assume(c_4, 'real');

    x1(t) = ((c_3 - c_2/4)*cos(t) + c_4*sin(t) - c_1/4*t*cos(t) ...
        - c_2/4*t*sin(t));
    x2(t) = ((c_4 - c_1/4)*cos(t) - c_3*sin(t) - c_2/4*t*cos(t) ...
        + c_1/4*t*sin(t));
    eqn1 = x1(0) == 1;
    eqn2 = x2(0) == 2;
    eqn3 = x1(t_f) == 0;
    eqn4 = x2(t_f) == 0;
```

```matlab

    res = solve([eqn1 eqn2 eqn3 eqn4], [c_1 c_2 c_3 c_4]);
end

function CE = plotter(tf,coeffs,fn1,fn2)
    c_1 = coeffs(1); c_2 = coeffs(2);
    c_3 = coeffs(3); c_4 = coeffs(4);
    lambda1 = @(t) c_1*cos(t) + c_2*sin(t);
    lambda2 = @(t) c_2*cos(t) - c_1*sin(t);
    u = @(t) -lambda2(t) / 2;
    u2 = @(t) (-(c_2*cos(t) - c_1*sin(t)) / 2).^2;
    uabs = @(t) abs(-(c_2*cos(t) - c_1*sin(t)) / 2);
    x1 = @(t) ((c_3 - c_2/4)*cos(t) + c_4*sin(t) ...
        - c_1/4*t.*cos(t) - c_2/4*t.*sin(t));
    x2 = @(t) ((c_4 - c_1/4)*cos(t) - c_3*sin(t) ...
        - c_2/4*t.*cos(t) + c_1/4*t.*sin(t));
    H = @(t) (u(t).^2 + lambda1(t) .* x2(t) ...
        + lambda2(t).*(-x1(t) + u(t)));

    tspan = linspace(0,tf,1000);
    x1sol = x1(tspan); x2sol = x2(tspan);
    fig = figure("Renderer","painters","Position",[60 60 950 800]);
        % x1 vs t
        subplot(2,2,1)
        plot(tspan, x1sol)
        title('$x_1$ over Time')
        xlabel('$t$')
        ylabel('$x_1$')
        grid on; grid minor; box on;
        % x2 vs t
        subplot(2,2,2)
        plot(tspan, x2sol)
        title('$x_2$ over Time')
        xlabel('$t$')
        ylabel('$x_2$')
        grid on; grid minor; box on;
        % x1 - x2
        subplot(2,2,3)
        plot(x1sol, x2sol,'-r',LineWidth=1.5)
        hold on;
        plot(x1sol(1),x2sol(1),'ok',MarkerSize=7)
        plot(x1sol(end),x2sol(end),'xk',MarkerSize=7)
        title('$x_1$ vs $x_2$')
        xlabel('$x_1$')
        ylabel('$x_2$')
```

```
        grid on; grid minor; box on; hold off;
        % Hamiltonian
        subplot(2,2,4)
        plot(tspan, H(tspan))
        title('Hamiltonian')
        xlabel('$t$')
        ylabel('$H$')
        ytickformat('%,.4f')
        grid on; grid minor; box on;
    saveas(fig, fn1);

    fig = figure("Renderer","painters","Position",[60 60 900 800]);
        % Plot costates
        subplot(2,1,1)
        plot(tspan, lambda1(tspan),'DisplayName','$\lambda_1$')
        hold on;
        plot(tspan, lambda2(tspan),'DisplayName','$\lambda_2$')
        title('$\lambda$ over Time')
        xlabel('$t$')
        ylabel('$\lambda_i$')
        legend('Location','best'); grid on; grid minor; box on; hold off;
        % Plot control
        subplot(2,1,2)
        plot(tspan, u(tspan))
        title('$u$ over Time')
        xlabel('$t$')
        ylabel('$u$')
        ytickformat('%,.4f')
        grid on; grid minor; box on;
    saveas(fig, fn2);

    % Control Efforts
    CE.int = sqrt(integral(u2,0,tf));
    CE.abs = integral(uabs,0,tf);
    CE.max = max(uabs(tspan));
end
```

9.5 Problem 6: MATLAB Code

```
% AE6511 Hw6 Problem 6 MATLAB code
% Tomoki Koike
clear all; close all; clc;  % housekeeping commands
set(groot, 'defaulttextinterpreter','latex');
```

```matlab
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%%
% Solve the differential equations when the control is u = 1
syms c c_1 c_2 c_3 c_4 c_5 c_6 t g
assume(c, {'real', 'positive'});
assume(c_1, 'real');
assume(c_2, 'real');
assume(c_3, 'real');
assume(c_4, 'real');
assume(c_5, 'real');
assume(c_6, 'real');
assume(t, 'real');
assume(g, {'real', 'positive'});
lambda1 = c_1;
lambda2(t) = -c_1 * t + c_2;
m(t) = -c*t + c_3;
lambda3 = int(lambda2 / m^2, t) + c_4
v = int(-g + 1/m, t) + c_5
h = int(v,t) + c_6
%%
clear;
syms t c g v_0 h_0 m_net m_0 t_s m_f T
assume(t, 'real');
assume(c, {'real', 'positive'});
assume(g, {'real', 'positive'});
assume(v_0, {'real', 'positive'});
assume(h_0, {'real', 'positive'});
assume(m_net, {'real', 'positive'});
assume(m_0, {'real', 'positive'});
assume(t_s, {'real', 'positive'});
assume(m_f, {'real', 'positive'});
assume(T, {'real', 'positive'});

% Before t_s
h_minus = -g/2*t_s^2 + v_0*t_s + h_0;
v_minus = -g*t_s + v_0;
m_minus = m_net + m_0;
% After t_s
c_3 = m_f + c*T;
c_5 = g*T + 1/c * log(c_3 - c*T);
c_6 = g*T^2/2 - c_5*T - 1/c*(T - T*log(c_3 - c*T)) - c_3/c^2 * log(c_3 - c*T);
h_plus = (-g/2*t_s^2 + c_5*t_s + 1/c*(t_s - T*log(c_3 - c*t_s)) ...
    + c_3/c^2 * log(c_3 - c*t_s) + c_6);
```

```
v_plus = -g*t_s - 1/c*log(c_3 - c*t_s) + c_5;
m_plus = -c*t_s + c_3;

% Equate the equations before the switching time and after
eqn1 = h_minus == h_plus;
eqn2 = v_minus == v_plus;
eqn3 = m_minus == m_plus;

% Solve for m_f and t_s
res =  solve([eqn2 eqn3], [t_s m_f], 'ReturnConditions',true)
%%
res.t_s
res.m_f
```

9.6 Problem 7: MATLAB Code

```
% AE6511 Hw6 Problem 7 MATLAB code
% Tomoki Koike
clear all; close all; clc;  % housekeeping commands
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
%%
% Define the matrices
syms s t alpha T
assume(alpha > 1);
R1 = -alpha;
R2 = 1;
A = 1;
B = 1;
x0 = 0;
xf = 0;
%%
A_tilde = -sqrt(1 - alpha);
W_tilde(t) = int(expm(A_tilde*s)*B*inv(R2)*B.'*expm(A_tilde.'*s),s,0,t)
%%
z0 = expm(A_tilde.'*T)*inv(W_tilde(T))*(expm(A_tilde.'*T)*x0 - xf)
%%
syms s(t)
dsolve(diff(s,t) == s^2 + 1)
%%
t = linspace(0,pi,100);
plot(t, 1 + tan(t - pi/4))
```