AE 551: Introduction to Optimal Control **(Due: 2020/06/26)**

Homework #10 Submission

Instructor: Min-Jea Tahk *Name:* Joshua Julian Damanik, *Student ID:* 20194701

The source code for this homework can be found at https://github.com/joshuadamanik/Homework-10
Cost Function

$$J = \int_0^{t_f} \frac{1}{2} u^2 dt \tag{1}$$

Dynamics

$$\dot{x} = v_x, \quad \dot{v_x} = 0 \tag{2}$$
$$\dot{y} = v_y, \quad \dot{v_y} = u \tag{3}$$

Initial Conditions:

$$x(0) = 0, \quad y(0) = 1000\ m, \quad v_x(0) = 250\ m/s, \quad v_y(0) = 0 \tag{4}$$

Terminal Constraints:

- CASE A :

$$x(t_f) = 5000\ m, \quad y(t_f) = 0 \tag{5}$$

- CASE B :

$$x(t_f) = 5000\ m, \quad y(t_f) = 0, \quad v_y(t_f) = 0 \tag{6}$$

Control Bounds:

$$|u| \leq 8\ m/s^2 \tag{7}$$

Methods : Your report should include all three methods described below.

1. Analytic Solution
   - Apply the Euler-Lagrange conditions. Ignore the control bounds if you can not solve the problem with bounded control by hand.
2. Hermite-Simpson Method
   - Include the control $|u| \leq 8\ m/s^2$
   - Use the Hermite-Simpson method for transcription. You can apply any parameter optimization tools but you need to clearly describe the method in the report.
3. Pseudo-Spectral Method
   - Include the control $|u| \leq 8\ m/s^2$
   - Use GPOPS II or your own code.

Compare the results of the three methods and summarize what you learn from this homework.

Problem 1: Analytic Solution

$$\min \quad J = \frac{1}{2}\int_{t_0}^{t_f} u^2 dt$$

s.t. $\dot{x} = v_x \qquad \dot{v}_x = 0$

$\qquad \dot{y} = v_y \qquad \dot{v}_y = u$

Initial conditions

$x(0) = 0 \qquad v_x(0) = 250$

$y(0) = 1000 \qquad v_y(0) = 0$

Terminal constraints

→ Case A : $x(t_f) = 5000 \qquad y(t_f) = 0$

→ Case B : $x(t_f) = 5000 \qquad y(t_f) = 0 \qquad v_y(t_f) = 0$

Control Bounds

$|u| \leq 8$

Euler-Lagrangian:

$H = \frac{1}{2}u^2 + \lambda_x v_x + \lambda_y v_y + \lambda_{v_y} u$

$\dot{\lambda}_x = -H_x = 0 \Rightarrow \lambda_x = c_1$

$\dot{\lambda}_y = -H_y = 0 \Rightarrow \lambda_y = c_2$

$\dot{\lambda}_{v_x} = -H_{v_x} = -\lambda_x \Rightarrow \lambda_{v_x} = -c_1 t + c_3$

$\dot{\lambda}_{v_y} = -H_{v_y} = -\lambda_y \Rightarrow \lambda_{v_y} = -c_2 t + c_3$

$\lambda_{v_{x_f}} = 0 \Rightarrow 20c_1 = c_3$

$\lambda_{v_{y_f}} = 0 \Rightarrow 20C_2 = C_3$

$\Downarrow$

$\lambda_{v_x} = (20-t)C_1$

$\lambda_{v_y} = (20-t)C_2$

For constrained input,

$H_u = u + \lambda_{v_y}$

At unsaturated condition, $u = -\lambda_{v_y}$

At saturated condition: $u = \begin{cases} 8, & \text{for } \lambda_{v_y} < -8 \\ -8, & \text{for } \lambda_{v_y} > 8 \end{cases} \Rightarrow$

For constrained input,

$$u = \begin{cases} 8, & \text{for } \lambda_{v_y} < -8 \\ -\lambda_{v_y}, & \text{for } -8 \leq \lambda_{v_y} \leq 8 \\ -8, & \text{for } \lambda_{v_y} > 8 \end{cases}$$

From dynamics,

$\dot{v}_x = 0 \Rightarrow v_x(t) = v_x(0) = 250$

$\dot{x} = v_x \Rightarrow x(t) = x(0) + v_x t = 250t$

$\Rightarrow$ For both case, A and B,

$x(t_f) = 250t_f = 5000$

$t_f = 20.$

Case A

$\Phi = \phi + \nu\psi = \nu_x(x-5000) + \nu_y y$

$\Phi_{v_y} = \lambda_{v_{yf}} = -20C_2 + C_3 \Rightarrow C_3 = 20C_2 \Rightarrow \lambda_{v_y} = (20-t)C_2$

Assume u never saturate, it must satisfy

$u = -\lambda_{v_y} \Rightarrow \dot{v}_y = u = (t-20)C_2 \Rightarrow v_y = v_y(0) + \left(\frac{1}{2}t^2 - 20t\right)C_2$

$-8 \le \lambda_{v_y} \le 8 \qquad\qquad = \left(\frac{1}{2}t^2 - 20t\right)C_2$

$\dot{y} = v_y \Rightarrow y = \left(\frac{1}{6}t^3 - 10t^2\right)C_2 + y(0)$

$= \left(\frac{1}{6}t^3 - 10t^2\right)C_2 + 1000$

At final time,

$y(20) = \left(\frac{8000}{6} - 4000\right)C_2 + 1000 = 0$

$C_2 = \frac{1000}{16000} \times 6 = \frac{3}{8}$

So, $\lambda_{v_y} = \frac{3}{8}(20-t)$

at $t \in [0, 20]$, $\lambda_{v_y} \in [0, 7.5]$

The λ_{v_y} satisfy inequality $-8 \le \lambda_{v_y} \le 8$ for $t \in [t_0, t_f]$

thus assumption $u = -\lambda_{v_y}$ holds.

$\therefore$ For case A, $u^* = \frac{3}{8}(t-20)$

$v_y^* = \frac{3}{16}t^2 - \frac{15}{2}t$

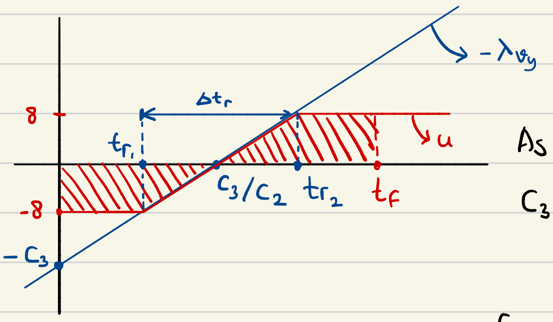
$y^* = \frac{1}{16}t^3 - \frac{15}{4}t^2 + 1000$

For $0 \le t \le 20$

Case B

$\dot{v}_y = u \Rightarrow v_y(t) = \int_0^t u(\tau)d\tau$

At final time, $v_y(t_f) = \int_0^{t_f} u(\tau)d\tau = 0$



As $\int_0^{t_f} u(\tau)d\tau = 0$, we can get relation

$C_3/C_2 = \frac{1}{2}t_f$

$C_3 = 10\,C_2$

$-\lambda_{v_y} = (t-10)\,C_2$

From the graph, we can derive
$$u = \begin{cases} -8, & \text{for } t \in \left[0, 10 - \frac{\Delta t_r}{2}\right] \\ (t-10)C_2, & \text{for } t \in \left[10 - \frac{\Delta t_r}{2}, 10 + \frac{\Delta t_r}{2}\right] \\ +8, & \text{for } t \in \left[10 + \frac{\Delta t_r}{2}, 20\right] \end{cases}$$

At $t = t_{r_1}$, $-8 = -\frac{\Delta t_r}{2}C_2$

$\Delta t_r = \frac{16}{C_2}$, $0 < \Delta t_r < 10 \Rightarrow t_{r_1} = 10 - \frac{8}{C_2}$, $t_{r_2} = 10 + \frac{8}{C_2}$

For $0 \le t \le t_{r_1}$, $v_y(t) = \int_0^t -8\,dt = -8t$

For $t_{r_1} \le t \le t_{r_2}$, $v_y(t) = v_y(t_{r_1}) + \int_{t_{r_1}}^{t_{r_2}} (t-10)c_2\,dt$

$= c_2\left(\frac{1}{2}t^2 - 10t + 50\right) + \frac{32}{c_2} - 80$

Because v_y is a continuous function,

$$v_y(t_{r_1}) = \frac{64}{c_2} - 80 = c_2\left(\frac{1}{2}\left(10 - \frac{8}{c_2}\right)^2 - 10\left(10 - \frac{8}{c_2}\right) + 50\right) + \frac{32}{c_2} - 80$$

$\Rightarrow c_2 = 1 \Rightarrow \Delta t_r = \frac{16}{c_2} = 16.$

But it contradicts with $0 < \Delta t_r < 10$

This means there is no feasible u that satisfy the boundary condition.

To solve the optimization, we need to ignore the control bounds.

Then we can solve for v_y and y

$\dot{v}_y = u = -\lambda_y = C_2 t - C_3 \Rightarrow v_y(t) = \frac{1}{2} C_2 t^2 - C_3 t$

At final time, $v_y(20) = 200C_2 - 20C_3 = 0$

$C_3 = 10 C_2 \Rightarrow v_y(t) = C_2\left(\frac{1}{2}t^2 - 10t\right)$

$\dot{y} = v_y \Rightarrow y(t) = 1000 + C_2\left(\frac{1}{6}t^3 - 5t^2\right)$

$y(20) = 1000 + C_2\left(8000/6 - 2000\right) = 0$

$C_2 = 3/2 \Rightarrow C_3 = 15$

Then $u^* = \frac{3}{2}t - 15$

$v_y^* = \frac{3}{4}t^2 - 15t$

$y^* = \frac{1}{4}t^3 - \frac{15}{2}t^2 + 1000$

for $0 \leq t \leq 20$

Problem 2: Hermite-Simpson Method

Looking at dynamic equations 2 and 3, the the control input only affect the vertical motion, thus we can simply ignore the horizontal equation at the optimization. However, we could obtain the final-time from the equation 5.

$$\begin{aligned} x(t_f) = v_x t_f = 250\ t_f &= 5000 \\ t_f &= 20\ s \end{aligned}$$

The dynamic equations are discritized using Hermite-Simpson method defined as follows.

$$v_{y_{k+1}} = v_k + \frac{h}{6}(u_k + 4u_c + u_{k-1}) \tag{8}$$

$$y_{k+1} = y_k + \frac{h}{6}(v_{y_k} + 4v_{y_c} + v_{y_{k-1}}) \tag{9}$$

with $h = 20/N$, N : number of time parameter. Because input control is a linear function, as proofed in problem 1, we can write the u_c as average value

$$u_c = \frac{1}{2}(u_k + u_{k-1}) \tag{10}$$

The value of v_{y_c} is approximated using cubic-spline as

$$v_{y_c} = \frac{1}{2}(v_{y_{k-1}} + v_{y_k}) + \frac{h}{8}(u_{k-1} - u_k) \tag{11}$$

So, we can rewrite the discrete dynamic equations as

$$y_{k+1} = y_k + \frac{h}{6}(3v_{y_k} + 3v_{y_{k-1}} + \frac{h}{2}(u_{k-1} - u_k)) \tag{12}$$

$$v_{y_{k+1}} = v_k + \frac{h}{2}(u_k + u_{k-1}) \tag{13}$$

To solve the optimization problem, define an optimization parameter

$$X = [y_1,\ y_2,\ \ldots,\ y_N,\ v_{y_1},\ v_{y_2},\ \ldots,\ v_{y_N},\ u_0,\ u_1,\ \ldots,\ u_{N-1}]^\intercal \quad \text{(3N variables)} \tag{14}$$

and the dynamic equations can be represented as a matrix equation

$$h(X) = AX - C = 0 \tag{15}$$

where

$$A = \left[\begin{array}{ccccc|ccccc|ccccc} 6 & 0 & \dots & 0 & 0 & -3h & 0 & \dots & 0 & 0 & -\frac{h}{2} & \frac{h}{2} & \dots & 0 & 0 \\ -6 & 6 & \dots & 0 & 0 & -3h & -3h & \dots & 0 & 0 & 0 & -\frac{h}{2} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 6 & 0 & 0 & 0 & \dots & -3h & -3h & 0 & 0 & \dots & \frac{h}{2} & 0 \\ 0 & 0 & \dots & -6 & 6 & 0 & 0 & \dots & 0 & -3h & 0 & 0 & \dots & -\frac{h}{2} & \frac{h}{2} \\ \hline 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & -\frac{h}{2} & \frac{h}{2} & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & -1 & 1 & \dots & 0 & 0 & 0 & -\frac{h}{2} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 & \dots & -\frac{h}{2} & -\frac{h}{2} \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & -1 & 1 & 0 & 0 & \dots & 0 & -h \end{array}\right], \quad C = \left[\begin{array}{c} 6y(0) \\ 0 \\ \vdots \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{array}\right] \tag{16}$$

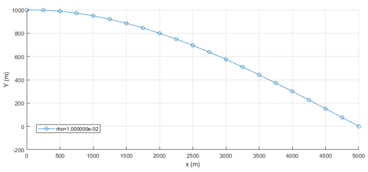
To inclue the final-state constraints, the matrix A and C is augmented with the dynamic equations 5 for case A or 6 for case B.

The optimization is done by using Penalty Functions of Augmented Lagrangian Methods. In this method, we defind an augmented lagrangian function

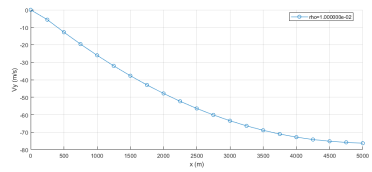
$$L_A(X, \lambda, \mu, \rho) = f(X) + \sum_{i=1}^{m} p_i(X, \mu_i, \rho) + \lambda^\intercal h(X) + \rho \sum_{i=1}^{l} h_i^2(X) \tag{17}$$

$$p_i(X, \mu_i, \rho) = \begin{cases} \mu_i g_i(x) + \rho {g_i}^2(x), & \text{if } g_i(x) \geq -\frac{\mu_i}{2\rho} \\ -\frac{\mu_k^2}{4\rho}, & \text{if } g_i(x) - \frac{\mu_i}{2\rho} \end{cases} \tag{18}$$

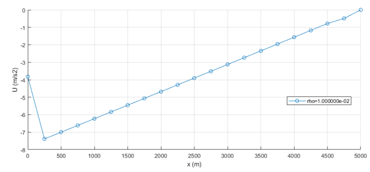
However, the control boundary results in infeasible solution, as proofed in analytical solution, the control boundary for case B is ignored. Thus, $\sum_{i=1}^{m} p_i(X, \mu_i, \rho) = 0$. Figure 1 shows the results of the optimization for case A, and figure 2 for case B.



(a) Position

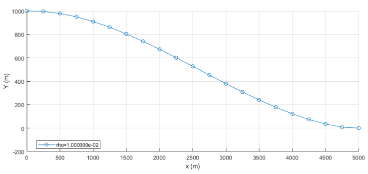


(b) Velocity

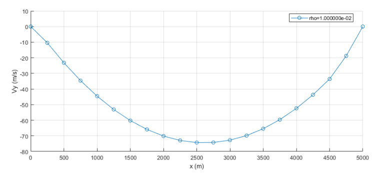


(c) Control

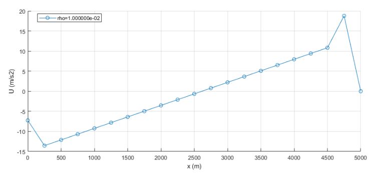
Figure 1: Results for case A



(a) Position



(b) Velocity

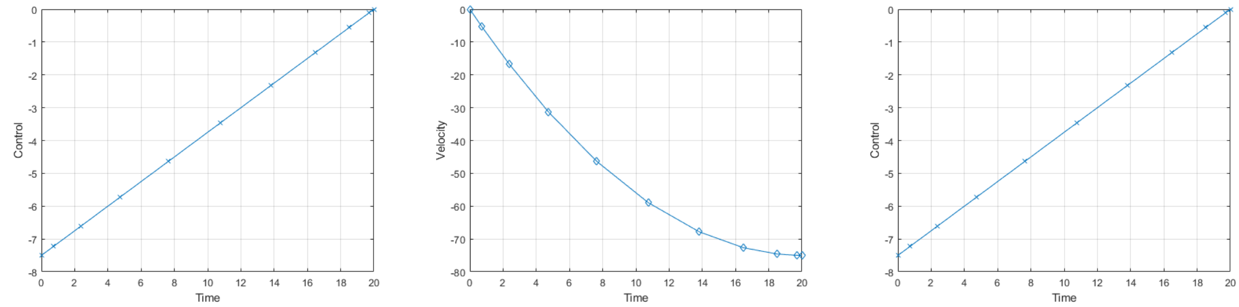


(c) Control

Figure 2: Results for case B

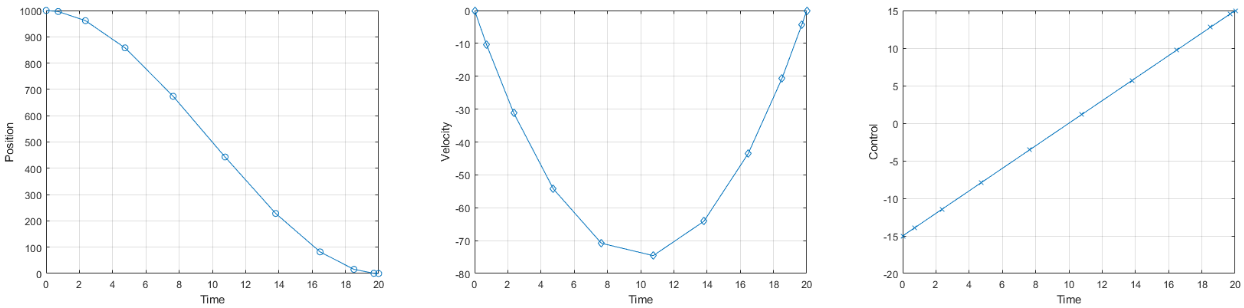
Problem 3: Pseudo-Spectral Method

For this section, the continuous dynamic functions is solved by pseudo-spectral method using GPOPS II software in MATLAB. The parameter used for GPOPS II is shown in figure 3. Figure 4 shows the result for case A and figure 5 shows the result for case B.



(a) Position (b) Velocity (c) Control

Figure 4: Results for case A



(a) Position (b) Velocity (c) Control

Figure 5: Results for case B

```
%--------------------------------------------------------------------------%
%----------------- Provide All Bounds for Problem ------------------------%
%--------------------------------------------------------------------------%


tmin = 0;       tmax = 1000;
xmin = -1e4;    xmax =  +1e4;
ymin = -1e4;    ymax =  +1e4;
vxmin = -1e4;   vxmax =  +1e4;
vymin = -1e4;   vymax =  +1e4;

umin_A  = -8;    umax_A = +8;
umin_B = -100;   umax_B = +100;

t0 = 0;

x0    = 0;      xf    = 5000;
y0    = 1000;   yf    = 0;
vx0   = 250;    vxf   = 250;
vy0   = 0;      vyf   = 0;


%--------------------------------------------------------------------------%
%------------------------ Setup for Problem Bounds ------------------------%
%--------------------------------------------------------------------------%
bounds.phase.initialtime.lower = t0;
bounds.phase.initialtime.upper = t0;
bounds.phase.finaltime.lower = tmin;
bounds.phase.finaltime.upper = tmax;
bounds.phase.initialstate.lower = [x0, y0, vx0, vy0];
bounds.phase.initialstate.upper = [x0, y0, vx0, vy0];
bounds.phase.state.lower = [xmin, xmin, vxmin, vymin];
bounds.phase.state.upper = [xmax, xmax, vxmax, vymax];
bounds.phase.integral.lower = 0;
bounds.phase.integral.upper = 1e6;

% FOR CASE A
% bounds.phase.finalstate.lower = [xf, yf, vxmin, vymin];
% bounds.phase.finalstate.upper = [xf, yf, vxmax, vymax];
% bounds.phase.control.lower = [uminA];
% bounds.phase.control.upper = [umaxA];

% FOR CASE B
bounds.phase.finalstate.lower = [xf, yf, vxmin, vyf];
bounds.phase.finalstate.upper = [xf, yf, vxmax, vyf];
bounds.phase.control.lower = [uminB];
bounds.phase.control.upper = [umaxB];
```

Figure 3: Parameter for GPOPS II