

Homework 7 - Optimal Control Systems

Erivelton Gualter dos Santos, 2703806

1 Zermelo's problem

- In order to reach the final condition, it was solved the zermelo's problem for the initial angle from 0 to 2π with a step size of 0.001. Therefore, the best initial condition angle was chosen according to the minimum euclidean distance to final position.
- Then, using the numerically solver “vpasolve” to find the solution for the following equation with the initial guess found previously, it reached the position in 5.4435 s.

$$x = -\frac{1}{2} \left[\sec \theta_f (\tan \theta_f - \tan \theta) - \tan \theta_f (\sec \theta_f - \sec \theta) + \ln \left(\frac{\tan \theta_f + \sec \theta_f}{\tan \theta + \sec \theta} \right) \right]$$

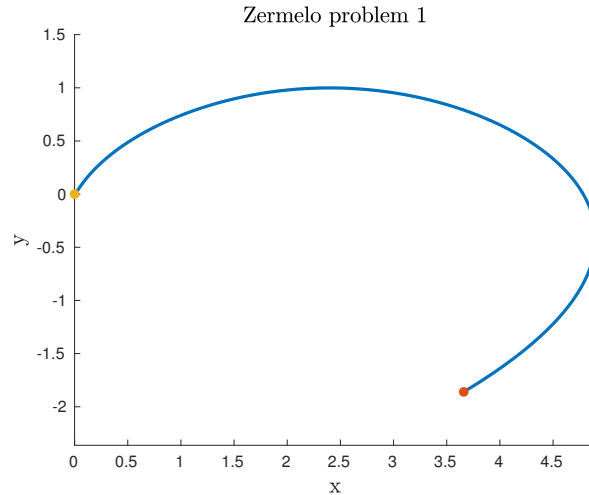


Figure 1: Zermelo's solution for $x(0) = 3.66$, $y(0) = -1.86$

2 Zermelo's problem

- As in the problem 1, it was performed an interaction guess for the angle from 0 to 2π with a step size of 0.001.
- Consequently, it was used the numerically solver to found θ and θ_f using the previously value as a guess.

$$x = -\frac{1}{2} \left[\csc \theta_f (\cot \theta_f - \cot \theta) - \cot \theta_f (\csc \theta_f - \csc \theta) + \ln \left(\frac{\cot \theta_f + \csc \theta_f}{\cot \theta + \csc \theta} \right) \right]$$

- The duration time to reach the final position is 4.0785 s.

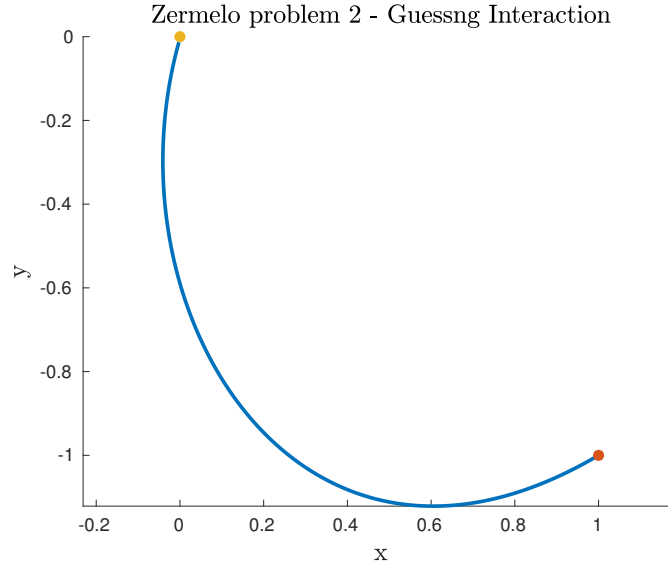


Figure 2: Zermelo's solution for $x(0) = 1, y(0) = -1$

3 Minimum-Drag Nose Shape

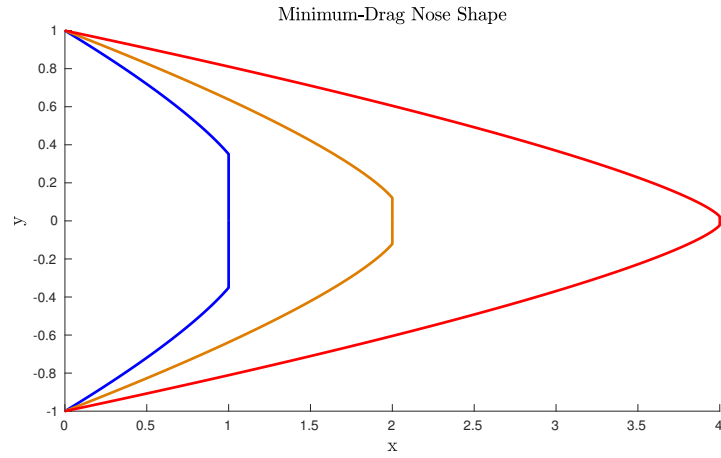


Figure 3: Minimum-Drag Nose Shape for $l=1,2$, and 4

4 Code

```
1 % Book: Optimal Control Theory: An introduction by Donald E. Kirk
2 %
3 % Erivelton Gualter, 03/19/2018
4
5 %% PROBLEM 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7 clear all; clc; close all
8
9 % Boundary Conditions
10 xo = 3.66;
11 yo = -1.86;
12 xf = 0;
13 yf = 0;
14
15 ts = 0.01;           % Sampling time
16 q = 0:0.001:2*pi;   % angle guess array
17 t = 0:ts:10;        % Array time
18
19 Z = zeros(length(t),3); % Initialize states
20 minimum = Inf;         % Variable to find the best guess
21
22 % Brush the angle array to find the best angle guess
23 for j=1:length(q)
24     Z(1,:) = [ xo yo q(j)]; % Initial state condition
25
```

```

26 % Euler Integration
27 for i=1:length(t)-1
28
29     % Load states
30     z1 = Z(i,1);
31     z2 = Z(i,2);
32     z3 = Z(i,3);
33
34     % Dot States
35     z1d = cos(z3)-z2;
36     z2d = sin(z3);
37     z3d = cos(z3)^2;
38
39     Zdot = [z1d z2d z3d];
40
41     % Integration
42     Z(i+1,:) = Z(i,:) + Zdot*ts;
43
44     % Euclidian distance of the final position to [xf yf]
45     best = norm(Z(i+1,1:2)-[xf yf]);
46     if best < minimum
47         minimum = best; % Update minimum flag
48
49         Zopt = Z; % Final States
50         topt = t(1:i); % time array
51         Zopt = Zopt(1:i,:);
52         tfinal = t(i);
53     end
54 end
55 end
56
57 % Find the angles using the guess condition
58 syms theta thetáf
59 sol = vpasolve([yo == sec(theta) - sec(thetáf), ...
60     xo == -(sec(thetáf)*(tan(thetáf)-tan(theta)) - ...
61     tan(theta)*(sec(thetáf)-sec(theta)) + ...
62     log((tan(thetáf)+sec(thetáf))/(tan(theta)+sec(theta)))/2] ...
63     , [theta thetáf], [Zopt(1,3) Zopt(end,3)]); ...
64     % [2.6270 1.2935]);
65 Z(1,:) = [xo yo double(sol.theta)]; % Initial condition
66 % Euler Integration Method
67 for i=1:length(t)-1
68
69     % Load states
70     z1 = Z(i,1);
71     z2 = Z(i,2);
72     z3 = Z(i,3);
73
74     % Dot States
75     z1d = cos(z3)-z2;
76     z2d = sin(z3);
77     z3d = cos(z3)^2;
78
79     Zdot = [z1d z2d z3d];
80

```

```

81     % Integration
82     Z(i+1,:) = Z(i,:) + Zdot*ts;
83 end
84
85 f1 = figure; hold on;
86 plot(Zopt(:,1), Zopt(:,2), 'LineWidth',2);
87 plot(xo, yo, '*', 'LineWidth',2);
88 plot(xf, yf, '*', 'LineWidth',2);
89 axis equal
90
91 title('Zermelo problem 1', 'Interpreter','Latex', 'FontSize',14);
92 xlabel('x', 'Interpreter','Latex', 'FontSize',14);
93 ylabel('y', 'Interpreter','Latex', 'FontSize',14);
94
95 saveFigureToPdf('f1', f1)
96
97 %% PROBLEM 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
98
99 clear all; clc;
100
101 % Boundary Conditions
102 xo = 1;
103 yo = -1;
104 xf = 0;
105 yf = 0;
106
107 ts = 0.01;          % Sampling time
108 topt = 0:ts:10;     % Array time
109 q = 0:0.001:2*pi;   % angle time
110
111 Z = zeros(length(topt),3);
112 minimum = Inf;
113
114 % Euler integration for q=0:2pi
115 for j=1:length(q)
116     Z(1,:) = [ xo yo q(j)]; % Load initial condition
117
118     for i=1:length(topt)-1
119
120         % States
121         z1 = Z(i,1);
122         z2 = Z(i,2);
123         z3 = Z(i,3);
124
125         % Derivative states
126         z1d = cos(z3);
127         z2d = sin(z3)-z1;
128         z3d = -sin(z3)^2;
129
130         Zdot = [z1d z2d z3d];
131
132         % Euler integration
133         Z(i+1,:) = Z(i,:) + Zdot*ts;
134
135         % Find best path to reach the final position
136         best = norm(Z(i+1,1:2)-[xf yf]);
137         if best < minimum

```

```

138         minimum = best;
139         Zopt = Z;
140         tfidx = i;
141         qo = q(j);
142     end
143 end
144 end
145
146 % reshape array
147 topt = topt(1:tfidx);
148 Zopt = Zopt(1:tfidx,:);
149
150 % Plot
151 figure; hold on
152 plot(Zopt(:,1), Zopt(:,2),'LineWidth',2);
153 plot(xo, yo, '*', 'LineWidth',2);
154 plot(xf, yf, '*', 'LineWidth',2);
155 axis equal
156 title('Zermelo problem 2 - Guessing Interaction', ...
157       'Interpreter','Latex', 'FontSize',14);
157 xlabel('x', 'Interpreter','Latex', 'FontSize',14);
158 ylabel('y', 'Interpreter','Latex', 'FontSize',14);
159
160 % Find the angles using the guess condition
161 syms theta thetaf
162 sol2 = vpasolve([yo == csc(theta) - csc(thetaf), ...
163                 xo == -(csc(thetaf)*(cot(thetaf)-cot(theta)) - ...
164                 cot(theta)*(csc(thetaf)-csc(theta)) + ...
165                 log((cot(thetaf)+csc(thetaf))/(cot(theta)+csc(theta)))/2] ...
166                 , [theta thetaf], [Zopt(1,3) Zopt(end,3)]);
167
168 Z(1,:) = [ xo yo double(sol2.theta)]; % Initial condition
169 for i=1:length(topt)-1
170
171     % States
172     z1 = Z(i,1);
173     z2 = Z(i,2);
174     z3 = Z(i,3);
175
176     % Derivative states
177     z1d = cos(z3);
178     z2d = sin(z3)-z1;
179     z3d = -sin(z3)^2;
180
181     Zdot = [z1d z2d z3d];
182
183     % Euler integration
184     Z(i+1,:) = Z(i,:) + Zdot*ts;
185
186     best = norm(Z(i+1,1:2)-[xf yf]);
187     if best < minimum
188         minimum = best;
189         Zopt = Z;
190         tfidx = i;
191         qo = q(j);
192     end

```

```

193 end
194
195 % Plots
196 figure; hold on
197 plot(Zopt(:,1), Zopt(:,2), 'LineWidth',2);
198 plot(xo, yo, '*', 'LineWidth',2);
199 plot(xf, yf, '*', 'LineWidth',2);
200 axis equal
201 title('Zermelo problem 2 – Finding guess angle directly', ...
       'Interpreter','Latex', 'FontSize',14);
202 xlabel('x', 'Interpreter','Latex', 'FontSize',14);
203 ylabel('y', 'Interpreter','Latex', 'FontSize',14);
204
205 %% PROBLEM 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
206
207 clear all, clc
208
209 % Parameters of the problem
210 a = 1;
211 l = 1; %1,2,4
212 dx = 0.01; % Sampling
213 x = 0:dx:l; % array x
214
215 % Initialization of u and r
216 u = zeros(size(x));
217 r = zeros(size(x));
218
219 % At x=0
220 syms uo rl un rn
221
222 % Solve r and u1 to the initial point
223 s1 = vpasolve([a == rl*(1+uo^2)^2/(4*uo^3), ...
224              l == rl*((3/(4*uo^4) + 1/uo^2 - 7/4 + ...
225              log(uo))/4), ...
226              [rl, uo], [0.3 0.5]]);
227
228 % Convert results to double
229 u(1) = double(s1.uo);
230 r(1) = a;
231 rl = double(s1.rl);
232
233 % Solve u and r for ΔX
234 % At x= dx*x
235 for i=2:length(x)-1
236     s2 = vpasolve([rn == rl*(1+un^2)^2 / (4*un^3), ...
237                 rl == (l-(i-1)*dx) / ((1/4)*(3/(4*un^4) + ...
238                 1/un^2 - 7/4 + log(un))), ...
239                 [un, rn], [u(i-1) r(i-1)]]);
240
241     r(i) = double(s2.rn); % Convert to double
242     u(i) = double(s2.un); % Convert to double
243 end
244
245 % Connect data
246 r(end) = rl;
247 x = [x l];
248 r = [r 0];

```

```
247
248 % Plots
249 hold on;
250 plot(x,r, x,-r,'b','LineWidth',2);
251 title('Minimum-Drag Nose Shape', 'Interpreter','Latex', ...
        'FontSize',14);
252 xlabel('x', 'Interpreter','Latex', 'FontSize',14);
253 ylabel('y', 'Interpreter','Latex', 'FontSize',14);
```

You can access the code at: <https://github.com/EriveltonGualter/EEC-744-Optimal-Control-Systems>