

Homework 9 - Optimal Control Systems

Erivelton Gualter dos Santos, 2703806

1 Steepest Descent Solution

The following figures (1, 2 and 3) are the results for the steepest descent method for the initial states $[0.5 \ 0]^T$, Initial control equal 1 and step size τ equal to 0.25. It reach the optimal trajectory in 48 interactions.

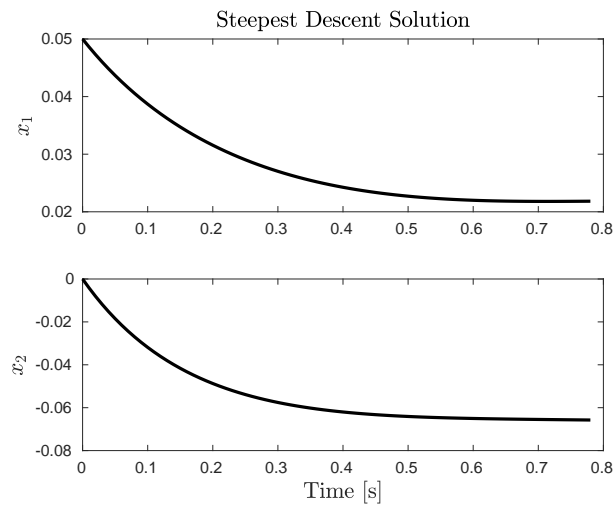


Figure 1: Optimal Control Trajectory

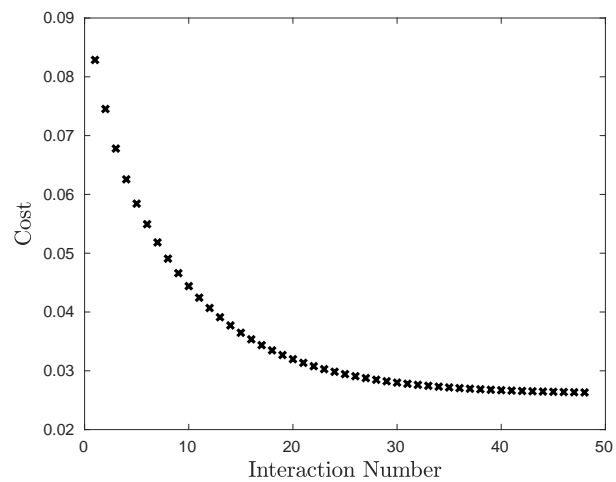


Figure 2: Performance Measure as a function of number of interactions

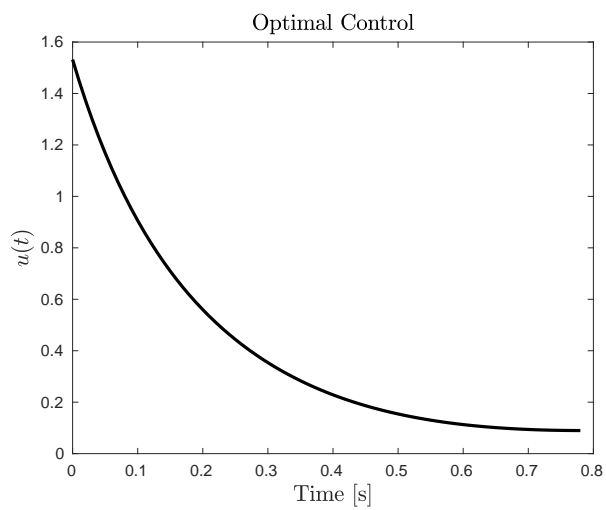


Figure 3: Optimal Control

Table 1: Number of interaction for different initial conditions				
Initial Control	Initial eps	Number of iterations required		Minimum value of J
1	0.25	48		0.026260622628662
1	0.5	25		0.026260622628662
0	0.25	19		0.026341012128041
0	0.1	42		0.026347117309669
0	0.01	386		0.026353420232887

From the table 1, it is clear that the initial condition makes a substantial difference relating to the number of interactions. Also, the magnitude of the gradient descent step size says how fast it will perform the new operation.

2 Variation of Extremal Solution

The following figures (4, 5 and 6) are the results for the Variation of Extremal method for the initial states $[0.5 \ 0]^T$ and $P0 = [1 \ 0.5]^T$. It reach the optimal trajectory in 5 interactions.

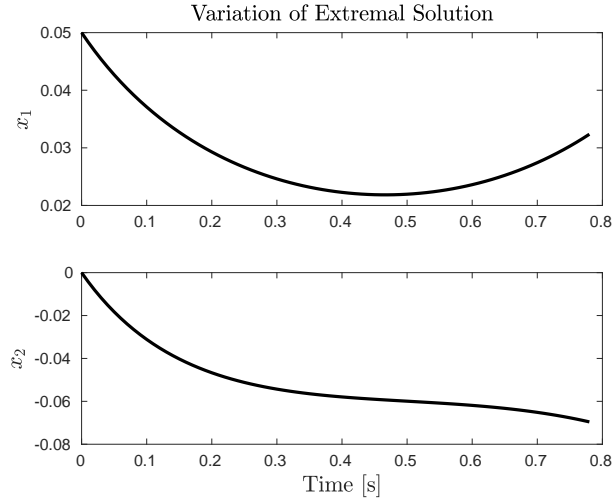


Figure 4: Optimal Control Trajectory

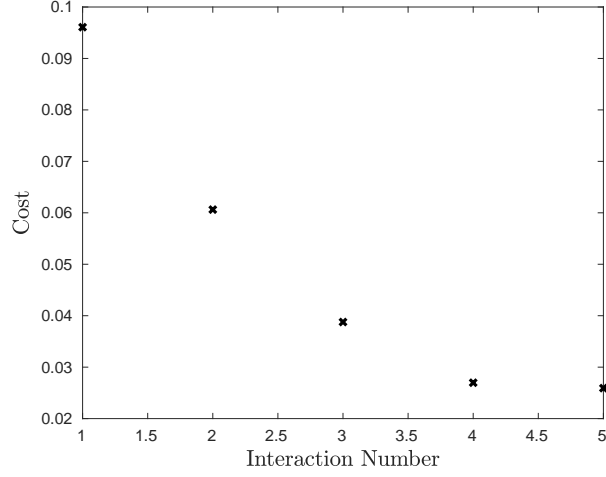


Figure 5: Performance Measure as a function of number of interactions

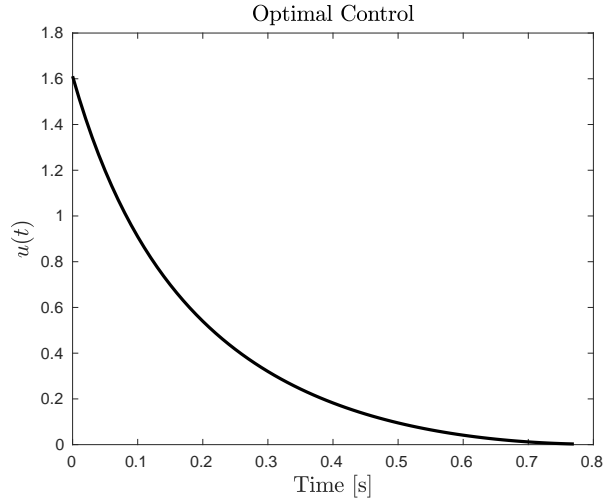


Figure 6: Optimal Control

Table 2: Number of interaction for different initial conditions			
P.1(0)	P.2(0)	Number of iterations required	Minimum value of J
1	0.5	5	0.025950108657380
2	1	33	0.025934192378058
2	2	37	0.025932988188449

Depending of the initial conditions, the simulation can reach the optimal solution really fast, or can take several hundreds interactions. Therefore, it must be choose carefully and based in the system.

```

1  % Book: Optimal Control Theory: An introduction by Donald E. Kirk
2  %
3  % Erivelton Gualter, 03/26/2018
4  % Problem 6.34
5
6  clc;
7  clear all;
8  close all;
9
10 % Simulation Parameters
11 tf = 0.78;
12 dt = 0.01; % time step
13 t = 0 : dt : tf;
14 N = length(t); % number of time steps
15 X0 = [0.05; 0];
16
17 % Parameters
18 R = 0.1;
19 eps = 0.25;
20
21 u = ones(size(t));
22
23 NormHu = inf;
24 JArr = [];
25
26 for iter = 1 : Inf
27     % Simulation system
28     X = X0;
29     for k=1 : length(t)-1
30         x1 = X(1,k);
31         x2 = X(2,k);
32
33         xd1 = -2*(x1+0.25) + (x2+0.5)*exp(25*x1/(x1+2)) ...
34             -(x1+0.25)*u(k);
35         xd2 = 0.5 - x2 - (x2+0.5)*exp(25*x1/(x1+2));
36
37         XDOT = [xd1; xd2];
38
39         X(:,k+1) = X(:,k) + XDOT*dt;
40     end
41
42     % Compute the costate
43     p = zeros(2, N);
44     p(1, N) = 0;
45     p(2, N) = 0;
46
47     for i = N-1 : -1 : 1
48         x1 = X(1,i+1);
49         x2 = X(2,i+1);
50         p1 = p(1,i+1);
51         p2 = p(2,i+1);

```

```

52         pDot(1, i) = -2*x1 + 2*p1 - ...
53             p1*(x2+0.5)*(50/(x1+2)^2)*exp(25*x1/(x1+2)) + ...
54             p1*u(i+1) + ...
55             p2*(x2+0.5)*(50/(x1+2)^2)*exp(25*x1/(x1+2));
56         pDot(2, i) = -2*x2 - p1*exp(25*x1/(x1+2)) + p2*(1 + ...
57             exp(25*x1/(x1+2)));
58     p(:, i) = p(:, i+1) - dt * pDot(:, i);
59 end
60 % Compute cost
61 J = dt*trapz(X(1,:).^2 + X(2,:).^2 + R*u.^2);
62 JArr = [JArr J];
63
64 % Compute the partial of the Hamiltonian with respect to the ...
65     control
66 for i=1:N
67     Hu(i) = 2*R*u(i)-p(1,i)*(X(1,i)+0.25);
68 end
69 NormHu = sqrt(dt * trapz(Hu.^2));
70 disp(['Iteration # ', num2str(iter), ', Hu = ', num2str(NormHu)]);
71 Optmail
72 if NormHu < 0.01
73     break;
74 end
75 u = u - eps * Hu;
76 end
77
78 %%
79 close all
80 f1 = figure;
81 subplot(211); plot(t, X(1,:), '-k', 'LineWidth', 2)
82     title('Steepest Descent Solution', 'Interpreter', 'Latex', ...
83         'FontSize', 14);
84     ylabel('$x_1$', 'Interpreter', 'Latex', 'FontSize', 14);
85 subplot(212); plot(t, X(2,:), '-k', 'LineWidth', 2)
86     ylabel('$x_2$', 'Interpreter', 'Latex', 'FontSize', 14);
87     xlabel('Time [s]', 'Interpreter', 'Latex', 'FontSize', 14);
88
89 f2 = figure;
90 plot(JArr, 'xk', 'LineWidth', 2);
91     ylabel('Cost', 'Interpreter', 'Latex', 'FontSize', 14);
92     xlabel('Interaction Number', 'Interpreter', 'Latex', ...
93         'FontSize', 14);
94
95 f3 = figure;
96 plot(t, u, '-k', 'LineWidth', 2);
97     title('Optimal Control', 'Interpreter', 'Latex', 'FontSize', 14);
98     ylabel('$u(t)$', 'Interpreter', 'Latex', 'FontSize', 14);
99     xlabel('Time [s]', 'Interpreter', 'Latex', 'FontSize', 14);
100
101 saveFigureToPdf('fig1', f1);
102 saveFigureToPdf('fig2', f2);
103 saveFigureToPdf('fig3', f3);
104
105 % Book: Optimal Control Theory: An introduction by Donald E. Kirk

```

```

104 %
105 % Erivelton Gualter, 03/26/2018
106 % Problem 6,34 and 6.35
107
108 clc;
109 clear all;
110 close all;
111
112 % Simulation Parameters
113 tf = 0.78;
114 dt = 0.01; % time step
115 t = 0 : dt : tf;
116 N = length(t); % number of time steps
117
118 % Parameters
119 R = 0.1;
120
121 P0 = [1; 0.5];
122 % P0=[1.0782; 0.1918];
123 X0 = [0.05; 0];
124
125 JArr = [];
126
127 for iter = 1 : Inf
128
129     % Simulatio system
130     X = X0;
131     P = P0;
132     for k=1 : N-1
133         x1 = X(1,k);
134         x2 = X(2,k);
135         p1 = P(1,k);
136         p2 = P(2,k);
137
138         u(k) = p1*(x1+0.25)/(2*R);
139
140         xd1 = -2*(x1+0.25) + (x2+0.5)*exp(25*x1/(x1+2)) ...
            -(x1+0.25)*u(k);
141         xd2 = 0.5 - x2 - (x2+0.5)*exp(25*x1/(x1+2));
142         pd1 = -2*x1 + 2*p1 - ...
            p1*(x2+0.5)*(50/(x1+2)^2)*exp(25*x1/(x1+2)) + ...
            p1*u(k) + ...
            p2*(x2+0.5)*(50/(x1+2)^2)*exp(25*x1/(x1+2));
143         pd2 = -2*x2 - p1*exp(25*x1/(x1+2)) + p2*(1 + ...
            exp(25*x1/(x1+2)));
144
145         XDOT = [xd1; xd2];
146         PDOT = [pd1; pd2];
147
148         X(:,k+1) = X(:,k) + XDOT*dt;
149         P(:,k+1) = P(:,k) + PDOT*dt;
150
151     end
152
153     J = dt*trapz(X(1,1:end-1).^2 + X(2,1:end-1).^2 + R*u.^2);
154     JArr = [JArr J];
155
156
157     p = P(:,end);

```

```

158     ErrorNorm = norm(p);
159     if ErrorNorm < 0.01
160         break
161     end
162     if (norm(P(1,end)) + norm(P(2,end))) < 1e-5
163         break;
164     end
165     disp(['Cost = ', num2str(J), ', Error norm = ', ...
166         num2str(ErrorNorm)]);
167
168     % integrate the Px and Pp matrices
169     Px = zeros(2,2);
170     Pp = eye(2);
171
172     for i = 1 : N
173         x1 = X(1,i);
174         x2 = X(2,i);
175
176         alpha = exp(25*x1/(x1+2));
177         d2Hdpdx = [-2+50*(x2+0.5)*alpha/(x1+2)^2-(x1+0.25)*p1/R, ...
178             alpha; ...
179             -50*(x2+0.5)*alpha/(x1+2)^2, -1-alpha];
180         d2Hdp2 = [-(x1+0.25)^2/(2*R), 0; 0, 0];
181         d2Hdx2 = ...
182             [-2+(p2-p1)*(100*(23-x1)*(x2+0.5)/(x1+2)^4)*alpha + ...
183             p1^2/(2*R), 50*(p2-p1)*alpha/(x1+2)^2; ...
184             50*(p2-p1)*alpha/(x1+2)^2, -2];
185         d2Hdxdp = d2Hdpdx';
186
187         Pxdot = d2Hdpdx * Px + d2Hdp2 * Pp;
188         Ppdot = d2Hdx2 * Px - d2Hdxdp * Pp;
189         Px = Px + Pxdot * dt;
190         Pp = Pp + Ppdot * dt;
191     end
192     P0 = P0 - inv(Pp) * p % updated p(0) guess
193 end
194
195 %%
196 close all
197 f4 = figure;
198 subplot(211); plot(t, X(1,:), '-k','LineWidth',2)
199 title('Variation of Extremal Solution', ...
200     'Interpreter','Latex', 'FontSize',14);
201 ylabel('$x_1$', 'Interpreter','Latex', 'FontSize',14);
202 subplot(212); plot(t, X(2,:), '-k','LineWidth',2)
203 ylabel('$x_2$', 'Interpreter','Latex', 'FontSize',14);
204 xlabel('Time [s]', 'Interpreter','Latex', 'FontSize',14);
205
206 f5 = figure;
207 plot(JArr,'xk','LineWidth',2);
208 ylabel('Cost', 'Interpreter','Latex', 'FontSize',14);
209 xlabel('Interaction Number', 'Interpreter','Latex', ...
210     'FontSize',14);
211
212 f6 = figure;
213 plot(t(1:end-1),u, '-k','LineWidth',2);

```



```
209     title('Optimal Control', 'Interpreter','Latex', 'FontSize',14);
210     ylabel('$u(t)$', 'Interpreter','Latex', 'FontSize',14);
211     xlabel('Time [s]', 'Interpreter','Latex', 'FontSize',14);
212
213     saveFigureToPdf('fig4',f4);
214     saveFigureToPdf('fig5',f5);
215     saveFigureToPdf('fig6',f6);
```

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