

EN.530.603 Applied Optimal Control

HW #1 Solutions

Gowtham Garimella

September 25, 2013

1. To find stationary points and determine whether they are maxima, minima or saddle points:
For stationary points, the gradient has to be zero.

(a)

$$\begin{aligned}L(x) &= (1 - x_1)^2 + 100(x_2 - x_1^2)^2 \\ \nabla L(x) &= \begin{bmatrix} \frac{\partial L}{\partial x_1} \\ \frac{\partial L}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2(1 - x_1)(-1) + 200(x_2 - x_1^2)(-2x_1) \\ 2 \times 100(x_2 - x_1^2) \end{bmatrix} = \mathbf{0}_{2 \times 1} \\ \Rightarrow x_2 &= x_1^2 \quad \& \quad -1 + x_1 = 0 \\ \Rightarrow x_1 &= 1 \quad \& \quad x_2 = x_1^2 = 1\end{aligned}$$

Thus the stationary point is (1,1). Now to characterize the stationary point we look at the Hessian of the function:

$$\begin{aligned}\nabla^2 L(x) &= \begin{bmatrix} 2(1 - 200x_2 + 600x_1^2) & 2(-200x_1) \\ 200(-2x_1) & 200 \end{bmatrix} \\ \nabla^2 L(x)|_{(1,1)} &= H = \begin{bmatrix} 802 & -400 \\ -400 & 200 \end{bmatrix} \\ eig(H) &= \{1001.6, 0.4\}\end{aligned}$$

Thus the Hessian is positive definite which implies the stationary point (1,1) is a **strict local minima**

(b)

$$\begin{aligned}L(u) &= (u - 1)(u + 2)(u - 5) \\ \nabla L(u) &= \frac{\partial L}{\partial u} = (u - 1)(u + 2) + (u - 1)(u - 5) + (u + 2)(u - 5) = 3u^2 - 8u - 7 = 0 \\ \Rightarrow u^* &= \{3.36, -0.69\}\end{aligned}$$

Now to characterize the stationary points we look at the Hessian:

$$\begin{aligned}\nabla^2 L(u) &= \frac{\partial^2 L}{\partial^2 u} = 6u - 8 \\ \nabla^2 L(u)|_{u=3.36} &= 6(3.36) - 8 = 12.16 \\ \nabla^2 L(u)|_{u=-0.694} &= 6(-0.694) - 8 = -12.16\end{aligned}$$

Clearly, point $u = 3.36$ is **strict local minima** and point $u = -0.694$ is **strict local maxima**

(c)

$$\begin{aligned}L(u) &= (u_1^2 + 3u_1 - 4)(u_2^2 - u_2 + 6) \\ \nabla L(u) &= \begin{bmatrix} \frac{\partial L}{\partial u_1} \\ \frac{\partial L}{\partial u_2} \end{bmatrix} = \begin{bmatrix} (2u_1 + 3)(u_2^2 - u_2 + 6) \\ (u_1^2 + 3u_1 - 4)(2u_2 - 1) \end{bmatrix} = \mathbf{0}_{2 \times 1} \\ \forall u_2 \in \mathbb{R} \quad u_2^2 - u_2 + 6 &> 0 \\ \Rightarrow u_1 &= -3/2 \\ \Rightarrow u^* &= \begin{bmatrix} -3/2 \\ 1/2 \end{bmatrix}\end{aligned}$$

Now that we found the stationary point let us characterize it by looking at the Hessian of the above objective function:

$$\begin{aligned}\nabla^2 L(u) &= \begin{bmatrix} 2(u_2^2 - u_2 + 6) & (2u_1 + 3)(2u_2 - 1) \\ (2u_1 + 3)(2u_2 - 1) & 2(u_1^2 + 3u_1 - 4) \end{bmatrix} \\ \nabla^2 L(u)|_{u=(-1.5, 0.5)} &= \begin{bmatrix} 11.5 & 0 \\ 0 & -12.5 \end{bmatrix} \\ \Rightarrow \text{eig}(\nabla^2 L(u)) &= \{11.5, -12.5\}\end{aligned}$$

Since the eigen values are both positive and negative at the stationary point $u = (-3/2, 1/2)$, the point is a **saddle point**.

2. To find the stationary points and determine the maxima, minima or saddle points:

(a) Define L' using lagrangian multipliers as $L' := L(x) + \lambda f(x)$ where $\lambda \in \mathbb{R}$ is a scalar. To

find the stationary points, we have to equate the gradient to zero:

$$\begin{aligned}
L'(x) &= \frac{1}{2}(x_1^2 + x_2^2 + x_3^2) + \lambda(x_1 + x_2 + x_3) \\
\frac{\partial L'}{\partial x_1} &= x_1 + \lambda = 0 \quad (\text{at stationary point}) \\
\Rightarrow \lambda &= -x_1^* \quad (x_1^* \text{ is the stationary point}) \\
\frac{\partial L'}{\partial x_2} &= x_2 + \lambda = 0 \\
\frac{\partial L'}{\partial x_3} &= x_3 + \lambda = 0
\end{aligned}$$

$$\begin{aligned}
\text{From constraint } f(x) \quad x_1 + x_2 + x_3 &= 0 \\
\Rightarrow x_2^* = x_3^* = x_1^* = \lambda &= 0
\end{aligned}$$

Thus the stationary point is (0,0,0). To characterize the stationary point we look at the Hessian of the L' . Since $\lambda = 0$, $L' = L$. Thus

$$\nabla^2 L' = \nabla^2 L = H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \text{eig}(H) = \{1, 1, 1\}$$

Since the eigen values are all positive, the Hessian is positive definite and hence the stationary point (0,0,0) is a **strict local minima**

- (b) As in above equation define the augmented cost function and let's equate the gradient to zero:

$$\begin{aligned}
L'(u) &= L(u) + \lambda f(u) = (u_1^2 + 3u_1 - 4)(u_2^2 - u_2 + 6) + \lambda(u_1 - 2u_2) \\
\frac{\partial L'}{\partial u_1} &= (2u_1 + 3)(u_2^2 - u_2 + 6) + \lambda = 0 \quad (\text{at stationary point}) \\
\frac{\partial L'}{\partial u_2} &= (u_1^2 + 3u_1 - 4)(2u_2 - 1) - 2\lambda = 0 \quad (\text{at stationary point}) \\
\frac{\partial L'}{\partial \lambda} &= u_1 - 2u_2 = 0 \\
\Rightarrow u_1 &= 2u_2 \quad \text{and} \\
\Rightarrow (u_1^2 + 3u_1 - 4)(2u_2 - 1) + 2(2u_1 + 3)(u_2^2 - u_2 + 6) &= 0 \\
\Rightarrow (2u_2^2 + 3u_2 - 2)(2u_2 - 1) + (4u_2 + 3)(u_2^2 - u_2 + 6) &= 0 \\
\Rightarrow 8u_2^3 + 3u_2^2 + 14u_2 + 20 &= 0
\end{aligned}$$

The above cubic equation yields only one real root i.e $u_2 = -1.03$ i.e $u = (-0.515, -1.03)$. To characterize this stationary point we look at the Hessian of L' :

$$\begin{aligned}
\nabla^2 L' &= \begin{bmatrix} 2(u_2^2 - u_2 + 6) & (2u_1 + 3)(2u_2 - 1) \\ (2u_1 + 3)(2u_2 - 1) & 2(u_1^2 + 3u_1 - 4) \end{bmatrix} \\
\nabla^2 L'|_{u=(-0.515, -1.03)} = H &= \begin{bmatrix} 16.182 & -6.028 \\ -6.028 & -10.56 \end{bmatrix} \\
\text{eig}(H) &= \{17.48, -11.86\}
\end{aligned}$$

Since the eigen values of the Hessian are positive and negative, the stationary point(-0.515,-1.03) is a **saddle point**.

3. To optimize the quadratic cost function with constraints given as:

$$\begin{aligned} L(x, u) &= \frac{1}{2}x^T Qx + \frac{1}{2}u^T Ru \\ f(x, u) &= Ax + Bu + c = 0 \quad x \in \mathbb{R}^n, c \in \mathbb{R}^m \end{aligned}$$

Using the Lagrangian multipliers method:

$$\begin{aligned} L'(x, u) &= \frac{1}{2}x^T Qx + \frac{1}{2}u^T Ru + \lambda^T(Ax + Bu + c) \\ \frac{\partial L'}{\partial x} &= x^T Q + \lambda^T(A) = \mathbf{0}_{1 \times n} \\ \frac{\partial L'}{\partial u} &= u^T R + \lambda^T(B) = \mathbf{0}_{1 \times m} \\ \frac{\partial L'}{\partial \lambda} &= Ax + Bu + c = 0 \end{aligned}$$

Since Q and R are both positive definite, they are invertible.

$$\begin{aligned} \Rightarrow x^T &= -\lambda^T A Q^{-1} \\ \Rightarrow u^T &= -\lambda^T B R^{-1} \\ Ax + Bu &= c \quad \equiv \left(A Q^{T^{-1}} A^T + B R^{T^{-1}} B^T \right) \lambda = c \end{aligned}$$

For such a λ to exist,

$$c \in \text{span}\{A Q^{T^{-1}} A^T + B R^{T^{-1}} B^T\}$$

Second order sufficient condition requires:

$$dy^T \nabla^2 L' dy = H = dy^T \begin{bmatrix} Q & \mathbf{0}_{n \times m} \\ \mathbf{0}_{m \times n} & R \end{bmatrix} dy > 0$$

Since Q and R are both positive definite,

$$\begin{bmatrix} x & u \end{bmatrix} \begin{bmatrix} Q & \mathbf{0}_{n \times m} \\ \mathbf{0}_{m \times n} & R \end{bmatrix} \begin{bmatrix} x^T \\ u^T \end{bmatrix} = x^T Qx + u^T Ru > 0$$

Thus the Hessian is already positive definite and hence there is no required second order sufficiency condition. Which implies that the necessary and sufficient condition for minimizing the objective function is as given below:

$$\begin{aligned} \{A Q^{T^{-1}} A^T + B R^{T^{-1}} B^T\} \lambda &= c \\ x &= -Q^{T^{-1}} A^T \lambda \\ u &= -R^{T^{-1}} B^T \lambda \end{aligned}$$

4. Matlab code for both the parts are given as follows:

```
1 function [] = Hw1( )
2 %HW1 Summary of this function goes here
3 % Detailed explanation goes here
4
5 %%%%%%%%%%%%%%% Part A %%%%%%%%%%%%%%%
6 %Variables
7 syms x1 x2;
8 x = [x1;x2];
9
10 %Choose function
11 L = (1-x1)^2 + 100*(x2-x1^2)^2;
12 %Gradient and Hessian
13 gradL = jacobian(L,x)';
14 hessL = jacobian(gradL,x)';
15
16 %Matlab handles for values:
17 Lnum = matlabFunction(L,'vars',{x});
18 gradLnum = matlabFunction(gradL,'vars',{x});
19 hessLnum = matlabFunction(hessL,'vars',{x});
20
21 disp(['L = ' , char(L)]);
22 disp(['gradient L = ']); disp(gradL);
23 disp(['hessian L = ']); disp(hessL);
24
25
26
27 disp('Gradient descent');
28 %Choose step size
29 step = 0.005;
30 disp(['step = ', num2str(step)]);
31 %Initialpoint
32 itpoints = zeros(2,1000);
33 x0 = [0;0];
34 itpoints(:,1) = x0;
35 disp('Starting point = '); disp(x0);
36 %Finding the stationary points:
37 i = 1;
38 gradcurrent = gradLnum(x0);
39 %figure,clf, hold on;
40 while ((i < 10000) && (norm(gradcurrent) > 1e-4))
41     gradcurrent = gradLnum(x0);
42     x0 = x0 - step*gradcurrent;
43     disp(['iteration: ', num2str(i)]);
44     %disp(['norm of gradient: ', num2str(norm(gradcurrent))]);
45     %disp('gradient');
46     %disp(gradcurrent);
47     %plot
48     %plot(i,norm(gradcurrent),'*');
49     %pause(0.05);
50     i = i+1;
51     itpoints(:,i) = x0;
52 end
```

```

53 disp(['Final point found after',num2str(i),'is:']);
54 disp(x0);
55 disp('Value at final point:');
56 disp(Lnum(x0));
57 disp('final gradient');
58 disp(gradLnum(x0));
59 disp('hessian @ final point');
60 disp(hessLnum(x0));
61 disp(eig(hessLnum(x0)));
62 figure;
63 ezsurf(L,[-2,2]);
64 hold on, plot3(itpoints(1,:),itpoints(2,:),Lnum(itpoints),'g*-');
65 figure;
66 ezcontour(L,[0,2],100);
67 hold on, plot(itpoints(1,:),itpoints(2,),'mx-');
68 disp('Please press any key to continue');
69 pause;
70
71 %%%%%%%%%%%%%%% Part B %%%%%%%%%%%%%%%
72 clear;
73 close all;
74 figure;
75
76 syms x u;
77 L = x^2 + 10*u^2;
78 f = x - 2*u + 3;
79
80 ezcontour(L,[-5,5],100);
81 hold on, ezplot(f,[-5,5]);
82
83
84 options = optimset([], 'GradObj', 'on', 'GradConstr', 'on');
85 options.Display = 'iter';
86 %options.PlotFcns = @optimplotx;
87 y0 = [0; 2];
88 objfun(y0);
89 [yfinal] = fmincon(@objfun,y0,[],[],[],[],[],[], ...
90                  @noncon,options);
91 disp('yfinal'); disp(yfinal);
92 figure;
93 %plot the function:
94 syms x u;
95 y = [x;u];
96
97 L = x^2 + 10*u^2;
98
99 ezsurf(L,[-2,2]);
100 hold on;
101 f = x - 2*u + 3;
102 ezplot(f,[-2,2]);
103 end
104
105
106 function [Lval,gradLval] = objfun(y1)
107

```

```

108     syms x u;
109     y = [x;u];
110
111     L = x^2 + 10*u^2;
112     gradL = jacobian(L,y)';
113
114     Lnum = matlabFunction(L, 'vars', {y});
115     Lval = Lnum(y1);
116     plot(y1(2),y1(1), 'b*-');%plot on same figure the points
117     if(nargout >1)
118         gradLnum = matlabFunction(gradL, 'vars', {y});
119         gradLval = gradLnum(y1);
120     end
121 end
122
123 function [c,ceq,gc,gceq] = noncon(y1)
124
125     syms x u;
126     y = [x;u];
127
128     f = x - 2*u + 3;
129     gradf = jacobian(f,y)';
130
131     fnum = matlabFunction(f, 'vars', {y});
132     c = fnum(y1);
133
134     if(nargout >1)
135         ceq = [];
136     end
137     if(nargout > 2)
138         gradfnum = matlabFunction(gradf, 'vars', {y});
139         gc = gradfnum(y1);
140         gceq = [];
141     end
142 end

```

- (a) The above solution converges to $\{x, u\} = \{1, 1\}$ for very small step sizes. If the step size is larger, the solution does not converge and even diverges.
- (b) The analytical solution for minimization following lagrangian method is as follows:

$$\begin{aligned}
 L'(x, u) &= x^2 + 10u^2 + \lambda(x - 2u + 3) \\
 \frac{\partial L'}{\partial x} &= 2x + \lambda = 0 \\
 \frac{\partial L'}{\partial u} &= 20u + 2\lambda = 0 \\
 \frac{\partial L'}{\partial \lambda} &= x - 2u + 3 = 0 \\
 \Rightarrow \frac{-\lambda}{2} - 2\frac{\lambda}{10} + 3 &= 0 \quad \Rightarrow \{\lambda, x, u\} = \left\{\frac{30}{7}, \frac{-15}{7}, \frac{-3}{7}\right\}
 \end{aligned}$$

The above solutions match with fmincon implementation from matlab.

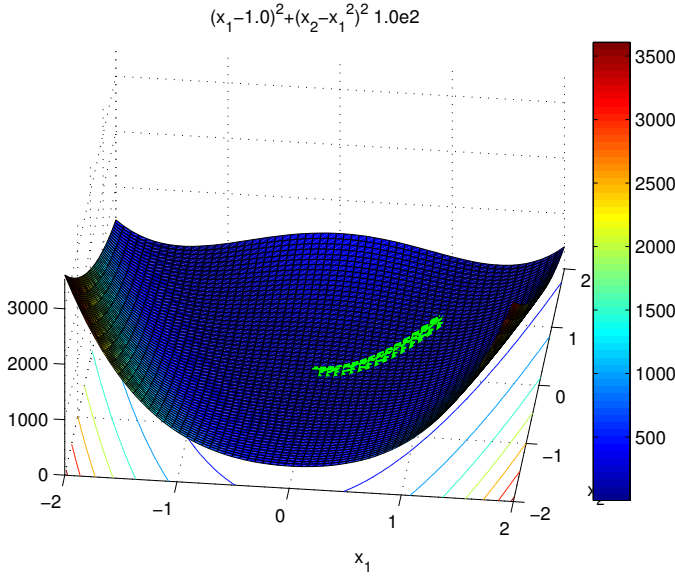


Figure 1: (a) plot of $L(x)$ and gradient descent

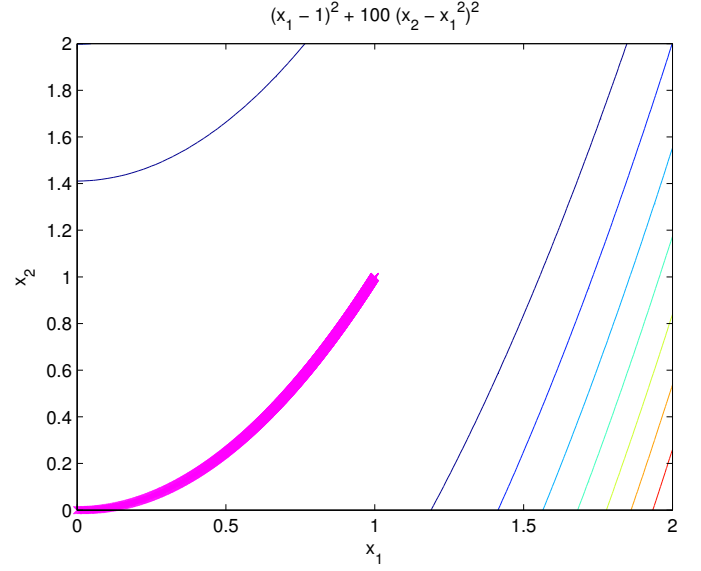


Figure 2: (a) contours and convergence of gradient descent to $(1,1)$

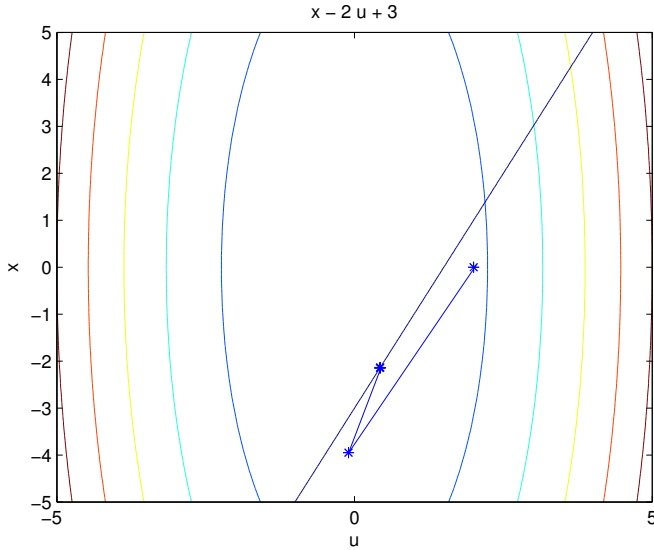


Figure 3: (b) contour plot of constrained minimization for $L = x^2 + 10u^2$

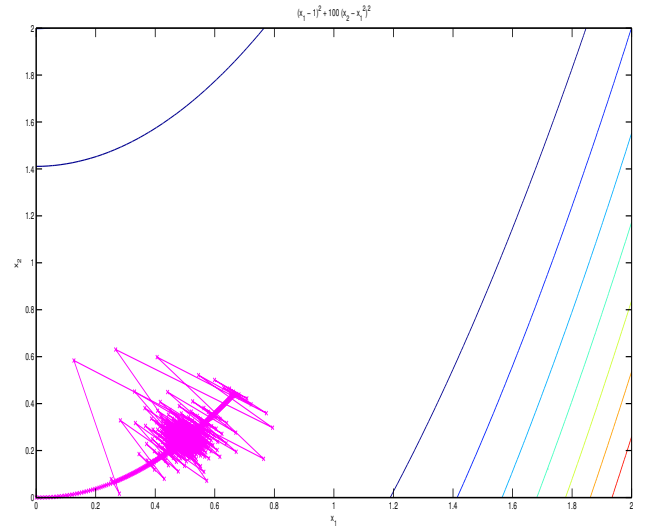


Figure 4: (a) divergence for step size 0.005

5. Minimization problem stated as:

$$f^k(d) = f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T \nabla^2 f(x^k) d \quad d \in \mathbb{R}^n$$

$$s.t \quad \|d\| \leq \gamma^k$$

If the constraint is active it is treated as an equality and following Lagrange multipliers

method:

$$\begin{aligned}
f(d) &= \|d\| - \gamma^k = \sqrt{d^T d} - \gamma^k \\
L'(d) &= f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T \nabla^2 f(x^k) d + \lambda(\sqrt{d^T d} - \gamma^k) \quad \lambda \geq 0 \\
\frac{\partial L'}{\partial d} &= \nabla f(x^k)^T + d^T \nabla^2 f(x^k) + \lambda\left(\frac{1}{2\sqrt{d^T d}} d^T\right) = 0 \\
\frac{\partial L'}{\partial \lambda} &= \sqrt{d^T d} - \gamma^k = 0 \\
\Rightarrow -\nabla f(x^k)^T &= d^T \left(\nabla^2 f(x^k) + \frac{\lambda}{\gamma^k} I \right) \\
\Rightarrow (\nabla^2 f(x^k) + \delta^k I) d &= -\nabla f(x^k) \quad \text{Since hessian and Identity are both symmetric}
\end{aligned}$$

Thus we have shown that the constrained optimization problem is equivalent to solving the above form of matrix equations for δ^k and d . We need to choose the right δ^k so that the value of $d^T d < \gamma^{k^2}$. A reasonable choice of δ^k is $\max\{-\text{eig}(\nabla^2 f(x^k)), 0\} + \epsilon$ $\epsilon \geq 0$. Since the eigen values of the incremented Hessian is the sum of original eigenvalues and δ^k . Such a choice will ensure all the eigen values of $(\nabla^2 f(x^k) + \delta^k I)$ at least ϵ . This satisfies the second order sufficiency condition for the Hessian. Also the value of ϵ should be found such that the value $\sqrt{d^T d} \leq \gamma^k$

6. Paper: **Arthur Bryson Optimal Control 1950-1985**

This paper describes the efforts of various authors in a 35 year period in the field of optimal control.

History of Calculus of Variations

Roots from calculus of variations

Optimal control can be regarded as an extension of calculus of variations. The paper describes some of the very early optimization problems posed by Galileo and Bernoulli(1600's) such as brachistochrone problem (shape of wire to minimize bead travel time); the drag nose shapes of projectile and other such problems. These were solved using Calculus of variations invented by Sir Newton and coinvented by Leibniz. Based on these works Euler and Lagrange(1700's) have proposed the first order necessary conditions for a stationary solution namely Euler-Lagrange equations. The first-order conditions for generalizations from calculus of variations have been discovered in 1750-1850 by the efforts of Legendre, Jacobi and Hamilton culminating in the Hamilton-Jacobi equations which formed basis for dynamic programming later on. The formal rigorous mathematical structure has finally been in place by efforts of Weirstrass and many others. Also at the same time many people were able to recognize the use of Calculus of Variations for engineering design.

Roots from Classical control theory

Optimal control in earlier days seen as a tool for adjusting the control system's performance to satisfactory and improve performance. This involved tuning the gains of the control systems and several criteria such as Nquist criterions etc have been proposed in the frequency domain. In 1960 Kalman proposed a method for measuring the performance of the system and controlling the feedback to the system known as LQR (Linear Quadratic Regulator).

Roots in Random Processes

Started with Einstein and extended by Wiener, Newton, Gould, Kalman etc culminating in Linear Quadratic Estimator(LQE). The Linear Quadratic Estimator and Regulator combined to form the Linear Quadratic Gaussian (LQG) compensator. Roots in Non-Linear Programming Kuhn and Tucker proposed necessary conditions for the optimization of a system under inequality constraints. This led to the non-linear programming codes which solve optimal trajectories etc under various constraints.

Effects of Technology on Optimal Control

The Non-Linear Programming was extremely benefited by the invent of computers. It provided the computational power to handle hundreds and thousands of variables in a Non-Linear Programming problem.

Important concepts developed in Optimal Control:

Dynamic Programming: This was invented by Bellman based on the Hamilton-Jacobi equations. It computes a set of extremal paths and an optimal return function along with the feedback required for such paths. These methods are not useful without approximating the system into a Linear Quadratic problem.

Maximum Principle: It is based on extending the Weirstrass's necessary condition to bounded control functions. It deals with find one extremum at a time and is different from Dynamic programming

Calulating NonLinear Optimal Trajectories First trajectories were proposed for rockets with limited fuel by Godard and were solved both analytically and numerically by various mathematicians. Later Gradient methods were proposed and coded by various people notably Hur and Bryson developed the MATLAB based gradient code. These problems have been applied to various real life situations in finding the right flight path to achieve a flight altitude in minimum time. Optimal control has also been used in finding optimal aerodynamic shapes etc.

Handling of various special Cases in OC

Inequality constraints: Inequalities in control variables were handled by introducing slack variables or penalty functions. State inequalities are harder to tackle and have been solved by Dreyfus and Speyer. *Singular problems:* The cases where the second derivative goes to zero is known as a singular solution. Approximate solutions have been found and are in general difficult to find. *Inverse Optimal Control:* These were required to produce a control history for a required output history. The finding of feasible output histories is hard and involves collocation methods and Non-Linear Programming concepts *Robust Optimal Control:* Linear Quadratic Gaussian controllers were not entirely robust to change in plant parameters. This came to light in using the controllers for space vehicles and new form of the problems known as H_∞ has been proposed for robustness to unmodelled high frequency and is an ongoing research field.

Thus Optimal Control has a long history in various fields such as Calculus of Variations, Control Theory, Non-Linear Programming and many other fields. It was applied to various real life problems during World War II for finding optimal trajectories and also in designing various engineering structures. Many new areas where Optimal control needs to be applied are opening up showing a bright future for the field.

Acknowledgements

I hereby declare that I have not discussed this homework with anyone. The solutions written here are my own work and from lecture notes and sample code provided by the professor. The paper summary contains lots of similarities with the original paper Arthur and Bryson “Optimal Control - 1950-1985”, IEEE Control Systems, 1996

Gowtham Garimella