EPCE . PG Final PCE6302 . Date June.08.2022

1. Given the cost function as

Find the to minimize using

* 1. Gradient Descent method (assume the tuning parameter , the initial point =(5,-5)

bY: The gradient of is

Hence

For

For

which is same to the previous value, i.e.,

* 1. Newton’s method (assume the initial point = (5,-5)

The Hessian is

which is independent of

Now the Newton’s method is

For

For

which is same to the previous value, i.e.,

1. (Bang-Bang Control)

Let the dynamic system as

And the input is bounded as

The control objective is to bring the states from any initial point ) to the origin in the minimum time , i.e., .

* 1. Define Hamiltonian and find its transversality condition.

Sol: The Hamiltonian is

The transversality condition is

Since ,

* 1. In the phase plane, draw the “switching curve”.

Sol: The phase-plane trajectories are first defined. Since

Eliminating the time variable is

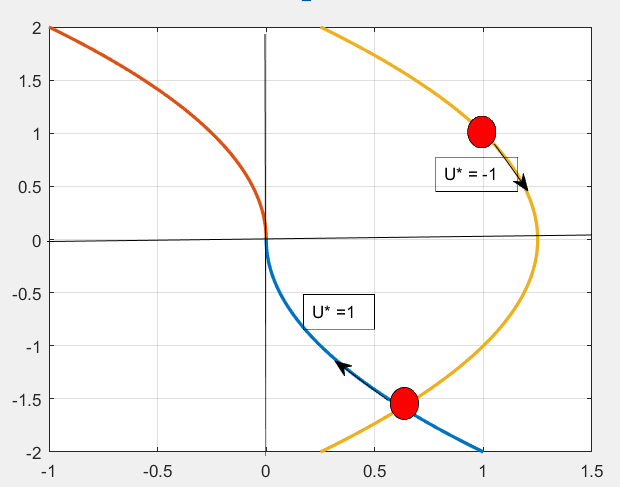
To find the switching curves, define then the (a) is

Since the optimal control is considering the optimality by “Pontryagin Principle”,

The switching curve is

* 1. Let the initial point as , Draw the optimal trajectory in the phase plane

Sol: Since , the equation (a) is



%% comment

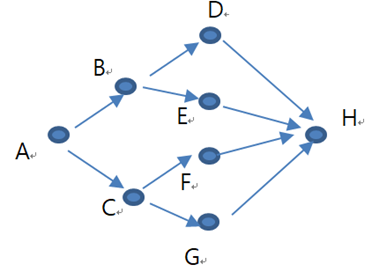
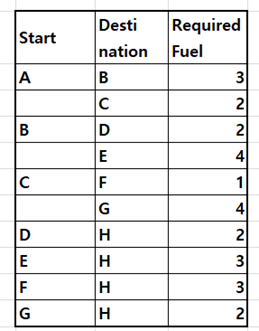
Unfortunately, there are no students to get the perfect point. Bang-bang is one of the famous

time optimal (minimum) problems. This case is seen in various real situations. In the car race, the winner is the fastest to get the destination, i.e., the minimum time. Pontryagin’s Principle should be used. Or, in machine learning, Lasso (the regularization is the absolute values of the features). And the solution is reasonable to understand. To go fastest, use the maximum acceleration and maximum de-acceleration.

Anyway, this problem is already in the mid-term exam,… %%%

1. (dynamic Programming) An aircraft Routing Problem , Ex. 6.1-1.

In the following graph, the nodes represent cities, the fuel for each path are in the table. Answer the questions



* 1. How many paths from “A” to “H”? Find the required fuel at each path form “A” to “H”. Find the minimum fuel path.

Sol:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Start |  |  | Destina tion | total fuel | min |
| A | **B** | **D** | **H** | 7 |  |
|  | B | E | H | 10 |  |
|  | C | F | H | 6 | min |
|  | C | G | H | 8 |  |

* Number of path:4
* Total fuel of each path:see table
* Minimum fuel path: ACFH  
  1. Using “Bellman Optimal Principle”, find the minimum path

Sol:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stage | State | Bellman | Value | Optimal |
| 3 | D | V(D) | 2 | **V(D)** |
|  | E | V(E ) | 3 | **V(E )** |
|  | F | V(F) | 3 | **V(F)** |
|  | G | V(G) | 2 | **V(G)** |
| 2 | B | 2+V(D) | 4 | **V(B)** |
|  |  | 4+V(E ) | 7 |  |
|  | C | 1+V(F) | 4 | **V(C )** |
|  |  | 4+V(G) | 6 |  |
| 1 | A | 3+V(B) | 7 |  |
|  |  | 2+V(C ) | 6 | **V(A)** |
|  |  |  |  | **Optimal** |
|  |  |  |  | **ACF H** |

%% Comments

1. The forward method looks simply compare to the backward.
2. The number of decision is 4 in the forward whereas 3 in backward.

(at the final stage, ‘3’, in backward, it is not included in the decision range)

1. If the number of nodes are increased, In backward, since the decision range is divided into subsets corresponding to “Bellman’s Principle”, the number of decision is reduced. This is efficient to find the optimal(**see google**)
2. Let us assume the start point is different, as In Forward, the optimal path should be recalculated, however in backward it is “CFH” or “B” is the starting point, “BDH”, so that the optimal path is dependent of the state, i.e., similar to **state feedback** !!

4. (Markov Decision Process)

Consider the following state transition diagram,

There are two control policies available , . Now the probability transition matrix with Control policy as

i.e.,

4.1 With the control policy , find the stead state probability transition matrix i.e.,

Sol:

Where are eigenvectors and eigenvalues respectively

Since

Which leads to

4.2 If the Reward as

Find the expectation Value as (be careful, the probability transition is the steady state transition)

Sol: **Here the equation above is missed the discount factor .**

**If , the Value equation is singular, i.e., the Value is not convergent**. !!

I should correct the formula

Now with the corrected formula , first with

Since the steady state transition probability matrix is

Rearrange it to get

And

So that

(c) and (d) are combined in matrix form

The linear system equation (e ) has no solution at ……

The solution of (e ) is

4.3 With the control policy , the transition is ergodic, the steady state transition probability is

Find the expectation Value as

Sol: The same as 4.2 with will be

The same method in 4.2 , in matrix form

So that

4.4 Given the present state , which control policy is to minimize the expectation Value?

Sol: Since

Hence

Similarly

Hence

%% comment:

The Value depends on the discount factor, but the optimal controller does not. %%%

4.5 In the point of optimal solution methods, what is the difference between dynamic programming of Problem 3 and MDP of Problem 4?

Sol:

1. In the dynamic programming, the optimal solution is determined by “backward” whereas

MDP is by “forward”

1. Backward is the off-line control so that the optimal value is computed before the actual action take place. Forward is the on-line control, which is computed as real time.
2. One of the bottlenecks over the optimal control problems is **“Backward”**. Since the system parameters are changed during the controlled process, every time at the different parameters the optimal control should be recalculated. (How ‘MPC’ solve this problem?) However, In the steady state MDP problems inherited probabilistic nature basically, it is allowed using **“Forward” which is guaranteed the Optimal control.**
3. To get the optimal using “Forward” there are some necessary conditions
4. The process is ergodic
5. One trick is to introduce “discounting factor” which is strictly less than 1 to insure the convergence of the optimal action.

-QED-