Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control
Institut für Dynamische Systeme und Regelungstechnik

Final Recitation Class **Dec 12th, 2012**

Dynamic Programming & Optimal Control (151-0563-00) **Mohanarajah G.**

Examples

Number of Examples: 7

Example 1

A burglar broke into a house and found $N \in \mathbb{Z}^+$ items[1]. Let $v_i > 0$ denote the value and $w_i \in \mathbb{Z}^+$ the weight of the i^{th} item. There is a limit, $W \in \mathbb{Z}^+$, on the total weight the burglar can carry (the total weight of all stolen items must be less than or equal to W) and obviously he wants to maximize the total value of the items that he can take.

a) Case A: A burglar who did not take the Dynamic Programming and Optimal Control class

> How many different combinations of items does the burglar have to consider to find the combination with the highest value that he can take?

b) Case B: A burglar who did take the Dynamic Programming and Optimal Control class

- b.1) Formulate the burglar's problem by defining the state space, control space, system dynamics, stage cost and terminal cost.
- b.2) State the Dynamic Programming Algorithm that is required to solve the burglar's problem.
- b.3) Using the Dynamic Programming Algorithm from b.2, solve the burglar's problem where $N = 3$, $W = 5$, $v_1 = 6$, $v_2 = 10$, $v_3 = 12$, $w_1 = 1$, $w_2 = 2$ and $w_3 = 3$.

c) Briefly compare Case A and Case B in terms of computational cost.

[1] $\mathbb{Z}^+$ denotes the set of positive integers

Solution 1

a) 2^N (Cardinality of the power set)

b) b.1) Let x_k, $k > 1$, be the total weight of the items the burglar has decided to take from the set of $\{1, .., k-1\}$ items and $x_1 = 0$. Furthermore, let $u_k \in \{0, 1\}$ be the binary decision variable that controls if item k goes into the burglar's bag or not.

State space $S_k \in [0, W] \subset \{0, \mathbb{Z}^+\}$, $k = 1, \dots, N+1$.
Control space U_k: if $x_k + w_k u_k > W$, $U_k = \{0\}$ else, $U_k = \{0, 1\}$, $\quad k = 1, \dots, N$.
System dynamics: $x_{k+1} = x_k + w_k u_k, \quad x_1 = 0, \quad k = 1, \dots, N$.
Stage cost: $g_k(x_k, u_k) = -v_k u_k, \quad k = 1, \dots, N$.
Terminal cost: $g_{N+1}(x_{N+1}) = 0, \quad \forall x_{N+1} \in S_{N+1}$.

b.2) Now applying the Dynamic Programming Algorithm (DPA) gives

$$\begin{aligned} J_{N+1}(x_{N+1}) &= g_{N+1}(x_{N+1}) = 0, \quad \forall x_{N+1} \in S_{N+1} \\ J_k(x_k) &= \min_{u_k}\{-v_k u_k + J_{k+1}(x_{k+1})\}, \quad k = 1, \dots, N. \end{aligned}$$

b.3) Applying the above DPA for $N = 3$, $W = 5$, $v_1 = 6$, $v_2 = 10$, $v_3 = 12$, $w_1 = 1$, $w_2 = 2$ and $w_3 = 3$, gives:
$J_4(x_4) = 0, \quad \forall x_4 \in \{0, 1, 2, 3, 4, 5\}$
$\underline{k = 3}$

$$\begin{aligned} J_3(0) &= \min\{-12 \cdot 0 + J_4(0 + 0 \cdot 3), -12 \cdot 1 + J_4(0 + 1 \cdot 3)\} = -12, \quad \mu_3(0) = 1 \\ J_3(1) &= \min\{-12 \cdot 0 + J_4(1 + 0 \cdot 3), -12 \cdot 1 + J_4(1 + 1 \cdot 3)\} = -12, \quad \mu_3(1) = 1 \\ J_3(2) &= \min\{-12 \cdot 0 + J_4(2 + 0 \cdot 3), -12 \cdot 1 + J_4(2 + 1 \cdot 3)\} = -12, \quad \underline{\mu_3(2) = 1} \\ J_3(3) &= \min\{-12 \cdot 0 + J_4(3 + 0 \cdot 3)\} = 0, \quad \mu_3(3) = 0 \\ J_3(4) &= \min\{-12 \cdot 0 + J_4(4 + 0 \cdot 3)\} = 0, \quad \mu_3(4) = 0 \\ J_3(5) &= \min\{-12 \cdot 0 + J_4(5 + 0 \cdot 3)\} = 0, \quad \mu_3(5) = 0 \end{aligned}$$

$\underline{k = 2}$

$$\begin{aligned} J_2(0) &= \min\{-10 \cdot 0 + J_3(0 + 0 \cdot 2), -10 \cdot 1 + J_3(0 + 1 \cdot 2)\} = -22, \quad \underline{\mu_2(0) = 1} \\ J_2(1) &= \min\{-10 \cdot 0 + J_3(1 + 0 \cdot 2), -10 \cdot 1 + J_3(1 + 1 \cdot 2)\} = -12, \quad \mu_2(1) = 0 \\ J_2(2) &= \min\{-10 \cdot 0 + J_3(2 + 0 \cdot 2), -10 \cdot 1 + J_3(2 + 1 \cdot 2)\} = -12, \quad \mu_2(2) = 0 \\ J_2(3) &= \min\{-10 \cdot 0 + J_3(3 + 0 \cdot 2) - 10 \cdot 1 + J_3(3 + 1 \cdot 2)\} = -10, \quad \mu_2(3) = 1 \\ J_2(4) &= \min\{-10 \cdot 0 + J_3(4 + 0 \cdot 2)\} = 0, \quad \mu_2(4) = 0 \\ J_2(5) &= \min\{-10 \cdot 0 + J_3(5 + 0 \cdot 2)\} = 0, \quad \mu_2(5) = 0 \end{aligned}$$

$\underline{k = 1}$

$$J_1(0) = \min\{-6 \cdot 0 + J_2(0 + 0 \cdot 1), -6 \cdot 1 + J_2(0 + 1 \cdot 1)\} = -22, \quad \underline{\mu_1(0) = 0}.$$

Therefore, optimal solution is to leave the first item and take the second and third items.

c) Case A has a time complexity $O(2^N)$ which is exponential in N and Case B has a time complexity $O(WN)$ which is linear in N. Therefore, for large N Case B (DPA) is computationally less expensive than Case A (Brute force approach).

Example 2

Consider the dynamic system

$$x_{k+1} = x_k + x_{k-1} + u_k + w_k\,, \quad k = 0, 1,$$

where the input u_k is restricted to be 1 or -1, and the disturbance w_k takes values 1 and -1 with equal probability.

Given $x_0 = 0$ and $x_{-1} = 0$, find the optimal control policy that minimizes

$$\sum_{k=0}^{2} (x_k - 1)^2$$

using the dynamic programming algorithm. Furthermore, calculate the corresponding optimal cost.

Solution 2

Introduce a new state variable $y_k = x_{k-1}$. This allows the system to be rewritten as

$$\tilde{x}_{k+1} := \begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + y_k + w_k + u_k \\ x_k \end{bmatrix},$$

with initial condition $\tilde{x}_0 = (0,0)$. Now, find all the states $\tilde{x}_k$ that can be reached from $\tilde{x}_0 = (0,0)$ with every possible u_k and w_k for $k = 1, 2$.

k=1 and $\tilde{x}_0 = (0,0)$:

$$\begin{aligned} \{u_0 = 1 \quad \text{and} \quad w_0 = 1\} \quad &\Rightarrow \quad \tilde{x}_1 = (2,0) \\ \{u_0 = 1 \quad \text{and} \quad w_0 = -1\} \quad \text{or} \quad \{u_0 = -1 \quad \text{and} \quad w_0 = 1\} \quad &\Rightarrow \quad \tilde{x}_1 = (0,0) \\ \{u_0 = -1 \quad \text{and} \quad w_0 = -1\} \quad &\Rightarrow \quad \tilde{x}_1 = (-2,0) \end{aligned}$$

k=2 and $\tilde{x}_1 = (2,0)$:

$$\begin{aligned} \{u_1 = 1 \quad \text{and} \quad w_1 = 1\} \quad &\Rightarrow \quad \tilde{x}_2 = (4,2) \\ \{u_1 = 1 \quad \text{and} \quad w_1 = -1\} \quad \text{or} \quad \{u_1 = -1 \quad \text{and} \quad w_1 = 1\} \quad &\Rightarrow \quad \tilde{x}_2 = (2,2) \\ \{u_1 = -1 \quad \text{and} \quad w_1 = -1\} \quad &\Rightarrow \quad \tilde{x}_2 = (0,2) \end{aligned}$$

k=2 and $\tilde{x}_1 = (0,0)$:

$$\begin{aligned} \{u_1 = 1 \quad \text{and} \quad w_1 = 1\} \quad &\Rightarrow \quad \tilde{x}_2 = (2,0) \\ \{u_1 = 1 \quad \text{and} \quad w_1 = -1\} \quad \text{or} \quad \{u_1 = -1 \quad \text{and} \quad w_1 = 1\} \quad &\Rightarrow \quad \tilde{x}_2 = (0,0) \\ \{u_1 = -1 \quad \text{and} \quad w_1 = -1\} \quad &\Rightarrow \quad \tilde{x}_2 = (-2,0) \end{aligned}$$

k=2 and $\tilde{x}_1 = (-2,0)$:

$$\begin{aligned} \{u_1 = 1 \quad \text{and} \quad w_1 = 1\} \quad &\Rightarrow \quad \tilde{x}_2 = (0,-2) \\ \{u_1 = 1 \quad \text{and} \quad w_1 = -1\} \quad \text{or} \quad \{u_1 = -1 \quad \text{and} \quad w_1 = 1\} \quad &\Rightarrow \quad \tilde{x}_2 = (-2,-2) \\ \{u_1 = -1 \quad \text{and} \quad w_1 = -1\} \quad &\Rightarrow \quad \tilde{x}_2 = (-4,-2) \end{aligned}$$

Calculate the terminal cost $J_2(\tilde{x}_2) = (x_2 - 1)^2$, for all possible states at $k = 2$.

$$\begin{aligned} &J_2((4,2)) = 9 & &J_2((2,0)) = 1 & &J_2((0,-2)) = 1 \\ &J_2((2,2)) = 1 & &J_2((0,0)) = 1 & &J_2((-2,-2)) = 9 \\ &J_2((0,2)) = 1 & &J_2((-2,0)) = 9 & &J_2((-4,-2)) = 25. \end{aligned}$$

Now, using the the dynamic programming algorithm,

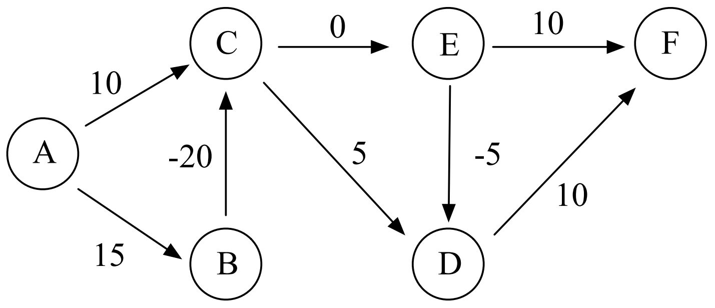
$$J_k(\tilde{x}_k) = J_k((x_k, x_{k-1})) = \min_{u_k \in \{-1,1\}} \underset{w_k}{E} \Big\{ (x_k - 1)^2 + J_{k+1}\big(\tilde{x}_{k+1}\big) \Big\},$$

for k = 1 and 0 gives

$$\begin{aligned} J_1((2,0)) &= 2 \quad \text{with} \quad \mu_1^*((2,0)) = -1, \\ J_1((0,0)) &= 2 \quad \text{with} \quad \mu_1^*((0,0)) = 1, \\ J_1((-2,0)) &= 14 \quad \text{with} \quad \mu_1^*((-2,0)) = 1, \quad \text{and} \\ J_0((0,0)) &= 3 \quad \text{with} \quad \mu_0^*((0,0)) = 1. \end{aligned}$$

Example 3

Consider the following transition graph.



a) Calculate the optimal cost to go and the shortest path from A to F using the dynamic programming algorithm.

b) Calculate the optimal cost to go and the shortest path from A to F using the label correcting algorithm. Use Breadth-first (First-in/First-out) search to determine at each iteration which node to remove from the `OPEN` bin.

Solve the problem by populating a table of the following form,

Iteration	Node exiting `OPEN`	`OPEN`	d_A	d_B	d_C	d_D	d_E	d_F
0	-	...						
...								

where the variable d_i denotes the length of the shortest path from node A to node i that has been found so far. State the resulting shortest path and the optimal cost-to-go.

c) How would the label correcting algorithm behave if a new arc from D to B is introduced with a weight of 10 ?

Solution 3

a) Dynamic Programming Algorithm:

$$\begin{aligned}
J(F) &= 0 \\
J(D) &= J(F) + 10 = 10 \quad \text{with} \quad \mu^*(D) = D \to F \\
J(E) &= \min\{10 + J(F), -5 + J(D)\} = 5 \quad \text{with} \quad \mu^*(E) = E \to D \\
J(C) &= \min\{0 + J(E), 5 + J(D)\} = 5 \quad \text{with} \quad \mu^*(E) = C \to E \\
J(B) &= -20 + J(C) = -15 \quad \text{with} \quad \mu^*(B) = B \to C \\
J(A) &= \min\{15 + J(B), 10 + J(C)\} = 0 \quad \text{with} \quad \mu^*(A) = A \to B
\end{aligned}$$

Therefore, the optimal cost to go from A to F is 0 and the shortest Path is $A \to B \to C \to E \to D \to F$

b) Label Correcting Algorithm:

Iteration	Node exiting `OPEN`	`OPEN`	d_A	d_B	d_C	d_D	d_E	d_F
0	–	A	0	∞	∞	∞	∞	∞
1	A	B,C	0	15	10	∞	∞	∞
2	B	C	0	15	-5	∞	∞	∞
3	C	D,E	0	15	-5	0	-5	∞
4	D	E,F	0	15	-5	0	-5	10
5	E	F,D	0	15	-5	-10	-5	5
6	F	D	0	15	-5	-10	-5	5
7	D	F	0	15	-5	-10	-5	0
8	F	–	0	15	-5	-10	-5	0

Therefore, optimal cost to go from A to F is 0 and the shortest Path is $A \to B \to C \to E \to D \to F$

c) The new arc will introduce a negative cycle ($D \to B \to C \to D$) and the label correcting algorithm will not terminate.

Example 4

Consider the stochastic shortest path problem shown in Figure 1

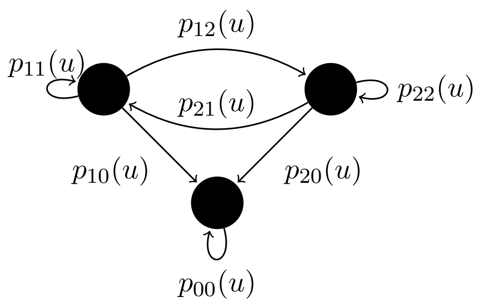


Figure 1: Transition graph of the stochastic shortest path problem

with the control sets

$$U(0) = \{a_0\}$$
$$U(1) = \{a_1, b_1, c_1\}$$
$$U(2) = \{a_2, b_2\},$$

the transition probabilities

$p_{10}(a_1) = 1/3$	$p_{20}(a_2) = 0$	$p_{00}(a_0) = 1$
$p_{11}(a_1) = 1/3$	$p_{21}(a_2) = 2/3$	
$p_{12}(a_1) = 1/3$	$p_{22}(a_2) = 1/3$	
$p_{10}(b_1) = 1/3$	$p_{20}(b_2) = 1/3$	
$p_{11}(b_1) = 2/3$	$p_{21}(b_2) = 1/3$	
$p_{12}(b_1) = 0$	$p_{22}(b_2) = 1/3$	
$p_{10}(c_1) = 0$		
$p_{11}(c_1) = 2/3$		
$p_{12}(c_1) = 1/3$		

and the cost function

$$g(i, \mu(i)) = 1 \quad i = 1, 2 \text{ and } \mu(i) \in U(i)$$
$$g(0, a_0) = 0.$$

a) Using policy iteration, perform 2 iterations for the given problem. Start with evaluating the initial policies $\mu^0(1) = a_1$ and $\mu^0(2) = a_2$.

b) Has the policy iteration converged after 2 iterations? If so, please explain why. If it has not converged, please explain the criterion that it would need to fulfill for convergence. Also, if it has not converged, can you comment on the possible outcome of iteration 3?

Solution 4

a) From the given information we can see that node 0 is the cost free terminal node. Therefore we do not need to consider it.

Iteration 1:
Policy evaluation:

$$\begin{aligned}
J^1(1) &= 1 + p_{11}(a_1)J^1(1) + p_{12}(a_1)J^1(2)\\
&= 1 + \frac{1}{3}J^1(1) + \frac{1}{3}J^1(2)\\
&\Rightarrow \underline{J^1(1) = \frac{3}{2} + \frac{1}{2}J^1(2)} \qquad (1)\\
J^1(2) &= 1 + p_{21}(a_2)J^1(1) + p_{22}(a_2)J^1(2)\\
&= 1 + \frac{2}{3}J^1(1) + \frac{1}{3}J^1(2)\\
&\overset{\text{using (1)}}{=} 1 + \frac{2}{3}\left(\frac{3}{2} + \frac{1}{2}J^1(2)\right) + \frac{1}{3}J^1(2)\\
&\underline{= 2 + \frac{2}{3}J^1(2)}\\
&\Rightarrow J^1(2) = 6,\ J^1(1) = 9/2
\end{aligned}$$

Policy improvement:

$$\begin{aligned}
\mu^1(1) &= \underset{u\in U(1)}{\operatorname{argmin}}\big[1 + p_{11}(a_1)J^1(1) + p_{12}(a_1)J^1(2),\\
&\qquad 1 + p_{11}(b_1)J^1(1) + p_{12}(b_1)J^1(2),\\
&\qquad 1 + p_{11}(c_1)J^1(1) + p_{12}(c_1)J^1(2)\big]\\
&= \underset{u\in U(1)}{\operatorname{argmin}}\,[1 + 1/3\cdot 9/2 + 1/3\cdot 6, 1 + 2/3\cdot 9/2 + 0\cdot 6, 1 + 2/3\cdot 9/2 + 1/3\cdot 6]\\
&= \underset{u\in U(1)}{\operatorname{argmin}}\,[9/2, 4, 6]\\
\mu^1(1) &= b_1\\
\mu^1(2) &= \underset{u\in U(2)}{\operatorname{argmin}}\big[1 + p_{21}(a_2)J^1(1) + p_{22}(a_2)J^1(2),\\
&\qquad 1 + p_{21}(b_2)J^1(1) + p_{22}(b_2)J^1(2)\big]\\
&= \underset{u\in U(2)}{\operatorname{argmin}}\,[1 + 2/3\cdot 9/2 + 1/3\cdot 6, 1 + 1/3\cdot 9/2 + 1/3\cdot 6]\\
&= \underset{u\in U(2)}{\operatorname{argmin}}\,[6, 9/2]\\
\mu^1(2) &= b_2
\end{aligned}$$

Iteration 2:

Policy evaluation:

$$\begin{aligned}
J^2(1) &= 1 + \frac{2}{3}J^2(1) + 0 * J^2(2) \\
&\Rightarrow J^2(1) = 3 \\
J^2(2) &= 1 + \frac{1}{3}J^2(1) + \frac{1}{3}J^2(2) = 2 + \frac{1}{3}J^2(2) \\
&\Rightarrow J^2(2) = 3
\end{aligned}$$

Policy improvement:

$$\begin{aligned}
\mu^2(1) &= \operatorname*{argmin}_{u \in U(1)} [1 + 1/3 \cdot 3 + 1/3 \cdot 3, 1 + 2/3 \cdot 3, 1 + 2/3 \cdot 3 + 1/3 \cdot 3] \\
&= \operatorname*{argmin}_{u \in U(1)} [3, 3, 4] \\
\mu^2(1) &= a_1 \text{ or } b_1 \\
\mu^2(2) &= \operatorname*{argmin}_{u \in U(2)} [1 + 2/3 \cdot 3 + 1/3 \cdot 3, 1 + 1/3 \cdot 3 + 1/3 \cdot 3] \\
&= \operatorname*{argmin}_{u \in U(2)} [4, 3] \\
\mu^2(2) &= b_2
\end{aligned}$$

b) Policy iteration has converged, if $J^k(i) = J^{k-1}(i)$ holds for all nodes i at iteration k. This is clearly not the case for $k = 2$. Therefore, policy iteration has not converged yet.

Even though policy iteration has not converged yet, we can still comment on the policy that the iteration will converge to.

Let's first pick $\mu^2(1) = b_1$ for iteration 3. In this case we apply the same combination of inputs as in iteration 2 and the transition probabilities in the policy evaluation step of iteration 3 will be the same. Therefore it will yield the same costs as in iteration 2, i.e., $J^2(i) = J^3(i)$ for all nodes i. Hence, policy iteration has converged to the policy $\mu(1) = b_1$ and $\mu(2) = b_2$.

Now, if we pick $\mu^2(1) = a_1$ for iteration 3 the only thing that we can tell about the convergence is that this choice will also eventually lead to convergence. But we cannot say after how many iterations or to which policy.

Additional information:
In fact, if you do iteration 3 with $\mu^2(1) = a_1$ you would also see that $J^2(i) = J^3(i)$ holds for all nodes i. Therefore, both choices for $\mu^2(1)$ are feasible outcomes of the policy iteration.

Example 5

Consider the one-dimensional linear system

$$\dot{x}(t) = ax(t) + bu(t), \quad x(0) = x_0 \in \mathbb{R}, \quad t \in [0, T],$$

where $u \in \mathbb{R}$ is the input, T is given, and $a, b \in \mathbb{R}$. The objective is to find an optimal policy that minimizes the quadratic cost

$$x^2(T) + \int_0^T (x^2(t) + u^2(t))dt.$$

Show that the cost-to-go function given by

$$J(t, x) = k(t)x^2,$$

where $k(t) \in \mathbb{R}$ is the solution to

$$\dot{k}(t) = -2a\ k(t) + b^2k^2(t) - 1, \quad k(T) = 1, \quad t \in [0, T],$$

is the *optimal* cost-to-go function and find the corresponding optimal policy as a function of $k(t)$ and x.

Solution 5

Let

$$G(t, x) := \min_{u \in U} \left[g(x, u) + \nabla_t V(t, x) + \nabla_x V^{\mathrm{T}}(t, x) f(x, u)\right]$$

denote the right hand side of the Hamilton-Jacobi-Bellman equation. Now substituting $g(x, u) = x^2 + u^2$ (stage cost) and $V(t, x) = J(t, x) = k(t)x^2$ (optimal cost-to-go function candidate) gives

$$G(t, x) = \min_{u \in U} \left[x^2 + u^2 + \dot{k}(t)x^2 + 2k(t)x(ax + bu)\right]. \tag{2}$$

The minimum is attained at a u for which the gradient with respect to u is zero; that is,

$$2u + 2bk(t)x = 0$$

or

$$u = -bk(t)x.$$

Substituting the minimizing value of u in (2) gives

$$\begin{aligned} G(t, x) &= x^2 + b^2k^2(t)x^2 + \dot{k}(t)x^2 + 2ak(t)x^2 - 2b^2k^2(t)x^2 \\ &= x^2(\dot{k}(t) + 2ak(t) - b^2k^2(t) + 1) \\ &= 0 \quad \forall x, t \in [0, T]. \end{aligned}$$

Note that the last equality results from the fact that $k(t)$ is a solution to

$$\dot{k}(t) = -2ak(t) + b^2k^2(t) - 1.$$

Furthermore,

$$V(T, x) = J(T, x) = k(T)x^2 = x^2 = h(x)(\text{terminal cost}), \quad \forall x.$$

Therefore, $J(t, x)$ is the optimal cost-to-go, i.e., $J^*(t, x) = J(t, x)$ and the corresponding optimal policy is given by

$$\mu^*(t, x) = -bk(t)x.$$

Example 6

Consider the dynamic system

$$\dot{x}(t) = x(t) + u(t)$$

with the initial state $x(0) = x_0$ and $t \in [0, T]$, $T \in \mathbb{R}^+$.

Use the Minimum Principle to find the optimal input $u^*(t)$ that minimizes the following cost function

$$\frac{1}{2}x^2(T) + \frac{1}{2}\int_0^T u^2(t)dt$$

and the corresponding optimal trajectory $x^*(t)$.

Solution 6

Applying the Minimum Principle:

The system equation is

$$\dot{x}(t) = x(t) + u(t). \tag{3}$$

The Hamiltonian is given by

$$\begin{aligned} H(x(t), u(t), p(t)) &= g(x(t), u(t)) + p(t) f(x(t), u(t)) \\ &= \frac{1}{2}u^2(t) + p(t)x(t) + p(t)u(t). \end{aligned}$$

The adjoint equation can be calculated as follows

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(x(t), u(t), p(t)) = -p(t),$$

with the boundary condition

$$p(T) = \nabla h(x(T)) = x(T).$$

Solving this differential equation leads to

$$p(t) = \zeta e^{-t}, \quad \zeta = x(T)e^{T} \tag{4}$$

The optimal input $u^*(t)$ is obtained by minimizing the Hamiltonian along the optimal trajectory

$$\begin{aligned} &u^*(t) = \underset{u}{\operatorname{argmin}}\left(\frac{1}{2}u^2(t) + p(t)x(t) + p(t)u(t)\right) \\ &\Rightarrow u^*(t) = -p(t). \end{aligned} \tag{5}$$

Now, (3), (4), and (5) give

$$\dot{x}(t) = x(t) - \zeta e^{-t}, \quad x(0) = x_0.$$

The general solution of the above inhomogeneous ordinary differential equation is the sum of the homogeneous solution $x_h(t) = \lambda e^t$ and a particular solution $x_p(t) = \frac{1}{2}\zeta e^{-t}$ giving

$$x(t) = x_h(t) + x_p(t) = \lambda e^t + \frac{1}{2}\zeta e^{-t}.$$

The above general solution with the initial condition $x(0) = x_0$, and $\zeta = x(T)e^T$ gives

$$\lambda = \frac{x_0}{1 + e^{2T}} \quad \text{and} \quad \zeta = \frac{2x_0 e^{2T}}{1 + e^{2T}}.$$

This implies

$$u^*(t) = -\frac{2x_0 e^{2T}}{1 + e^{2T}} e^{-t}$$

and

$$x^*(t) = \frac{x_0}{1 + e^{2T}} e^t + \frac{x_0 e^{2T}}{1 + e^{2T}} e^{-t}.$$

Example 7

Consider the system of water reservoirs shown in Figure 2

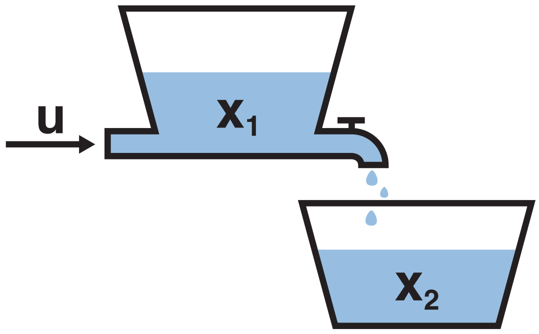


Figure 2: water reservoir system

where x_i, $i = 1, 2$ is the water volume of reservoir i and u is the flow rate of water into reservoir 1 via an external pump. Water exits reservoir 1 at a rate x_1 and enters reservoir 2.

a) Taking flow rate u as the control, find the system equations of the water reservoir system shown in Figure 2.

b) Assuming the external pump is unidirectional and has a maximum inflow rate of 2, i.e. $u \in U_b = [0, 2]$ compute the time optimal maneuver to fill each of the two empty reservoirs with 1 unit of water by applying the Minimum Principle. (When x_2 reaches 1 unit of water the valve between reservoir 1 and 2 will be closed to prevent more water flowing into reservoir 2.)

c) If the external pump is bidirectional and has a maximum flow rate of 2, i.e. $u \in U_c = [-2, 2]$, will the optimal maneuver to fill the two empty reservoirs with one unit of water be faster, slower or take the same amount of time as in the case with the unidirectional pump? Explain the reasons for your answer.

Solution 7

a)

$$\begin{aligned}\dot{x}_1(t) &= -x_1(t) + u(t)\\ \dot{x}_2(t) &= x_1(t)\end{aligned}$$

b)

- The boundary conditions for the system are $x_1(0) = x_2(0) = 0$ and $x_1(T) = x_2(T) = 1$.

- The objective to minimize is
$$T = \int_0^T 1\ dt.$$

- The Hamiltonian is
$$H(x(t), u(t), p(t)) = 1 - p_1(t)x_1(t) + p_1(t)u(t) + p_2(t)x_1(t).$$

- Adjoint equations
$$\begin{aligned}&\dot{p}_2 = 0 \ \Rightarrow p_2(t) = c\\ &\dot{p}_1 = p_1 - p_2 = p_1 - c \ \Rightarrow p_1(t) = \xi e^t + c \rightarrow \text{max. one zero crossing}\end{aligned}$$

- If $u^*(t)$is the optimal control and $x^*(t)$ is the optimal state trajectory, then the necessary condition for optimality is
$$u^*(t) = \underset{u\in U}{\operatorname{argmin}}\ H(x^*(t), u, p(t)).$$

- Since the Hamiltonian is linear in u, u will always be on a boundary of U:
$$u^*(t) = \begin{cases} 0 & \text{if } p_1(t) \geq 0\\ 2 & \text{if } p_1(t) < 0\end{cases}$$

- Since both containers are initially empty we start at $u = 2$. We know that we have at most one zero crossing of p_1. Therefore we will have to apply $u = 2$ until we have enough water in the reservoirs.
$$x_1(T_b) + x_2(T_b) = 2 = \int_0^{T_b} u(t)\ dt = \int_0^{t_{switch}} 2\ dt + \int_{t_{switch}}^{T_b} 0\ dt = 2t_{switch}$$
$\Rightarrow$ The optimal solution is to run the pump at $u = 2$ for 1 time unit and then switch it off and wait until enough water has run down into reservoir 2.

c) First of all, since the control set U_b is a subset of the control set U_c the maneuver cannot be slower because we can apply the solution obtained in b). To show that the maneuver using U_c is faster than the maneuver using U_b we can use a proof by contradiction:

Assume that the solution obtained in b) with U_b is also an optimal solution for c) with U_c. Then the solution from b) also needs to be a minimizer of the Hamiltonian given U_c. Since the adjoint equations and the Hamiltonian do not change, p_1 still has at most one zero crossing and the Hamiltonian is still linear in u. Therefore we know that the optimal solution is on the boundary of U_c and switches at most one time. But the solution obtained in b) is not on the boundary of U_c and therefore is not a minimizer for the Hamiltonian anymore. So the solution using U_c must be faster than the solution using U_b.