6 Lecture 6. Final step of the proof of MP and a start of DP

6.1 The proof of the maximum principle (finally!)

In our previous lecture, we started proving the maximum principle for the Mayer problem

$$\dot{x} = f(x, u)$$

with cost

$$J = \varphi(x(t_f))$$

under the constraint $u(t) \in U$, $x(t_f) \in M$. The basic tool for the proof is the method of tent. To that end, we defined the following tents:

$$\begin{aligned}
&\Omega_0 = \{x_1\} \cup \{x : \varphi(x) < \varphi(x_1)\} \\
&\Omega_1 : \text{ reachable set from } x_0 \\
&\Omega_2 = M\text{: the terminal manifold}
\end{aligned}$$

where $x_1 := x_*(t_f)$. Let $u_*(t)$, $x_*(t)$, $0 \le t \le t_f$ be an optimal process. Then we claimed that

$$\Omega_0 \cap \Omega_1 \cap \Omega_2 = \{x_1\}. \tag{1}$$

This condition, by Lemma 2, implies separability of tents of the three sets. Therefore, it suffices to find the tents of the three sets. Denote K_i as the tents of Ω_i at x_1. The tents K_0 and K_2 can be easily computed:

$$\begin{aligned}
K_0 &= \{x \in \mathbb{R}^n : \nabla\varphi(x_1)(x - x_1) \le 0\} \\
K_2 &= T_{x_1}\Omega_2
\end{aligned}$$

(note that Ω_2 is a fixed manifold).

Therefore, our problem boils down to finding a tent of Ω_1 at x_1: K_1. By definition, a tent is only a convex subcone of the tangent cone of Ω_1 at x_0. We should try to find a tent as big as possible, since the bigger the tent, the more necessary information it conveys. For that, we introduced *needle variation* of u_* for small $\varepsilon > 0$:

$$u_\varepsilon(t) = \begin{cases} w, & t \in (\tau - \varepsilon, \tau] \\ u_*(t), & \text{otherwise} \end{cases}$$

where $w \in U$ is some constant, see Figure 1.

Then we showed that

$$v(t_f) = \left.\frac{\partial x_\varepsilon(t_f)}{\partial \varepsilon}\right|_{\varepsilon=0+}$$

is a vector in the tangent cone of Ω_1, which we call a deviation vector, where $v(t)$ is a solution to the following linear ODE:

$$\begin{cases} \dot{v} = \dfrac{\partial f}{\partial x}(x_*(t), u_*(t))v, \quad \forall t \in [\tau, t_f] \\ v(\tau) = f(x_*(\tau), w) - f(x_*(\tau), u_*(\tau)). \end{cases}$$

Then we obtained as least one vector in the tangent cone of Ω_1. Now, repeating the perturbation at different time instants with different perturbation w, we can obtain many deviation vectors. Say $v_1(t_f), \cdots, v_r(t_f)$ are some deviation vectors obtained through needle variations at time instants $\tau_1 < \cdots < \tau_r$ with inputs $w_1, \cdots, w_r$. Then we showed the combined needle variation in Figure 2 generates the deviation vector

$$\sum_{i=1}^{r} k_i v_i(t_f) = \left.\frac{\partial x(t_f, u_{\varepsilon,k})}{\partial \varepsilon}\right|_{\varepsilon=0+}$$

If we define K_1 to be the set of all deviation vectors of such form, i.e.,

$$K_1 = \left\{ \sum_{i=1}^{r} k_i v_i(t_f) \;\middle|\; \begin{array}{l} \exists r \in \mathbb{Z}_+,\ \tau_i \in [0, t_f),\ w_i \in U,\ k_i \ge 0, \\ v_i(t_f) \text{ the deviation vector obtained from needle} \\ \text{variation at} \tau_i \text{ with spike } w_i \end{array} \right\}$$

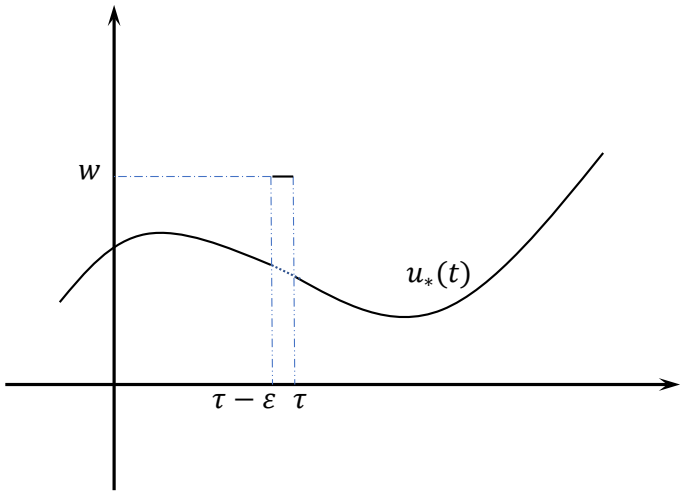

Figure 1: Needle variation.

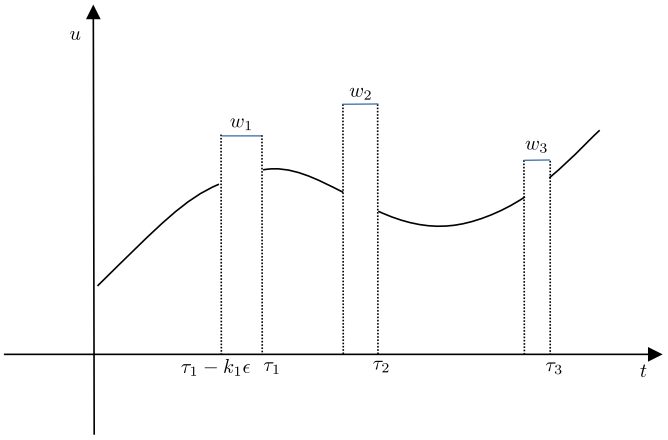

Figure 2: Combined needle variation

then we obtain a tent of Ω_1 at x_1.

Condition (1) implies that K_0, K_1, K_2 are separable. Invoking Lemma 1 and Lemma 2, we deduce that there exist three vectors a_i, at least one of which is nonzero, satisfying

$$a_i^\top v \le 0, \quad v \in K_i,\ i = 0, 1, 2 \tag{2}$$

and

$$a_0 + a_1 + a_2 = 0. \tag{3}$$

For a_0, since K_0 is a half space, $a_0^\top v \le 0$ for $v \in K_0$ implies $a_0 = \lambda \nabla\varphi(x_1)^\top$ for some constant $\lambda \ge 0$.

For a_1, we have $a_1^\top v(t_f) \le 0$ for any deviation vector $v(t_f)$. We emphasize that a_1 does not depend on specific needle variation. Now we introduce a small trick: if we are able to construct some function $p : [0, t_f] \to \mathbb{R}^n$ such that $p(t)^\top v(t) \equiv$ constant with $p(t_f) = a_1$, then we obtain immediately $p(t)^\top v(t) = a_1^\top v(t_f) \le 0$ for all $t \in [0, t_f]$. In other words, we propagate the inequality at the end point to the previous time instants. This is easy, let us recall the following simple fact:

Lemma 1. *Consider two linear ODE*

$$\dot{x} = A(t)x$$
$$\dot{p} = -A(t)^\top p$$

where $x, p \in \mathbb{R}^n$. *Then* $p(t)^\top x(t) = p(t')^\top x(t')$ *for any* $t, t' \in \mathbb{R}$.

Now since $A(t) = \frac{\partial f}{\partial x}(x_*(t), u_*(t))$, we have

$$\dot{p} = -\left[\frac{\partial f}{\partial x}(x_*(t), u_*(t)\right]^\top p$$

with terminal condition $p(t_f) = a_1$.

Therefore, if v is a deviation vector obtained by needle variation at time τ with spike w, then $v(\tau) = f(x_*(\tau), w) - f(x_*(\tau), u_*(\tau))$. Thus at $t = \tau$, $p(\tau)^\top [f(x_*(\tau), w) - f(x_*(\tau), u_*(\tau))] \le 0$ or

$$p(\tau)^\top f(x_*(\tau), u_*(\tau)) \ge p(\tau)^\top f(x_*(\tau), w)) \tag{4}$$

For convenience, define

$$H(x, u, p) := p^\top f(x, u)$$

which is the Hamiltonian associated with the system. Now that the spike can be any $w \in U$ and $t \in [0, t_f)$, it follows from (4) that

$$H(x_*(t), u_*(t), p(t)) = \max_{u \in U} H(x_*(t), u, p(t)), \quad \forall t \in [0, t_f). \tag{5}$$

Now, for any $t_1 > \tau$,

$$H(x_*(\tau), u_*(\tau), p(\tau)) - H(x_*(t_1), u_*(t_1), p(t_1)) \le H(x_*(\tau), u_*(\tau), p(\tau)) - H(x_*(t_1), w, p(t_1))$$

for any $w \in U$. In particular, take $w = u_*(\tau)$, we have

$$H(x_*(\tau), u_*(\tau), p(\tau)) - H(x_*(t_1), u_*(t_1), p(t_1)) \le H(x_*(\tau), u_*(\tau), p(\tau)) - H(x_*(t_1), u_*(\tau), p(t_1))$$

Since H is C^1 in x and u, we have

$$\begin{aligned} H(x_*(\tau), u_*(\tau), p(\tau)) - H(x_*(t_1), u_*(t_1), p(t_1)) &\le -H_x \dot{x}_*(\tau) + H_p \dot{p}(\tau) + o(|t_1 - \tau|) \\ &= o(|t_1 - \tau|) \end{aligned}$$

A reverse direction inequality can also be established. Therefore, $H(x_*(\tau), u_*(\tau), p(\tau))$ is differentiable at τ, and more precisely, its derivative vanishes, which happens only when H is constant. Thus we can improve (5) to

$$H(x_*(t), u_*(t), p(t)) = \max_{u \in U} H(x_*(t), u, p(t)) = \text{constant}, \quad \forall t \in [0, t_f). \tag{6}$$

This is the maximum principle that we have been looking for! Except two things: the interval $[0, t_f)$ doesn't include the endpoint t_f and the function p hasn't been determined yet. The first issue can be fixed if everything is continuous in the above formula, which is indeed true as long as we have shown p is, since f, x_* and u_* are continuous as assumed.

Now recall the equation of p, we can rewrite it as

$$\dot{p} = -H_x^\top (x_*, u_*, p) \tag{7}$$

with terminal state $p(t_f) = a_1$ which is exactly the costate equation! However, a_1 is something we need to determine. Recall that

$$\begin{aligned} K_0 &= \{x \in \mathbb{R}^n : \nabla \varphi(x_1)(x - x_1) \le 0\} \\ K_2 &= T_{x_1} \Omega_2 \end{aligned}$$

and $a_0 + a_1 + a_2 = 0$. For a_2, since K_2 is a sub-manifold, $a_2 \perp K_2$. It follows from (3) that (recall $a_1 = p(t_f)$):

$$\lambda \nabla \varphi(x_*(t_f))^\top + p(t_f) \perp T_{x_*(t_f)} M \tag{8}$$

for some constant $\lambda \ge 0$. If $\lambda > 0$, then it is equivalent to $p(t_f) + \nabla \varphi(x_*(t_f)) \perp T_{x_*(t_f)} M$ by changing a_1 to λa_1. As in many textbooks, we ignore the pathological case $\lambda = 0$.

Up to now, we have proven the maximum principle for the Mayer problem under the assumption that u_* is continuous. We can safely extend to the case when u is only piecewise continuous (more generally, measurable is enough) and modify the maximum principle to hold almost everywhere. Summarizing, we have proved the following.

Theorem 1. *Suppose that the Mayer form optimal control problem admits a piecese-continuous optimal law $u_*(\cdot)$ with corresponding trajectory $x_*(\cdot)$. Then there is a solution to the costate equation (7), such that the triple $(x_*(t), u_*(t), p(t))$ satisfies the maximum principle (6) for almost all t (all t on the interval on which $u_*(\cdot)$ is continuous) and the transversality condition (8).*

Discussions

- We have so far considered the optimal control problem under the condition that t_f is fixed. It can be easily extended to the case of free terminal time: it is obvious that all the necessary conditions of Theorem 1 still need to hold. The mere difference is that now one can also make the variation of the terminal time. For example, consider a needle variation at τ, let $v(t_f)$ be the corresponding deviation vector. Fix some $\mu > 0$, since $x_\epsilon(t_f + \epsilon\mu) \in \Omega_1$, $\left.\frac{\partial x_\varepsilon(t_f+\varepsilon\mu)}{\partial \varepsilon}\right|_{0+}$ must also lie in the tangent cone of Ω_1.
- As we said, the problem in Bolza form can be reduced to Mayer form, so in fact we have proved the general form of the maximum principle. We left the derivation as an exercise.

6.2 Dynamic programming

In the rest of the time, we will start learning another approach to optimal control, i.e., dynamic programming. The good news is that dynamic programming is much easier to understand than the maximum principle. In particular, if you work with discrete time control systems, the only knowledge you need to know is high school algebra, maybe some linear algebra if you deal with multi-dimensional systems. The bad news is that for some problems, especially for continuous time systems, the dynamic programming is quite hard to solve, where normally you have to solve a partial differential equations. This is not the case for the maximum principle. We will come back to this issue later.

6.2.1 Shortest path problem

To understand dynamic programming, perhaps it is best to start with the shortest path problem. The following directed graph (Figure 3) shows some possible paths connecting a starting point F to a target T. The number on each arrow indicates the cost walking from one node to the other, and the total cost is the sum of the costs of all moves. The objective is to find the path connecting F to T which has the minimal cost.

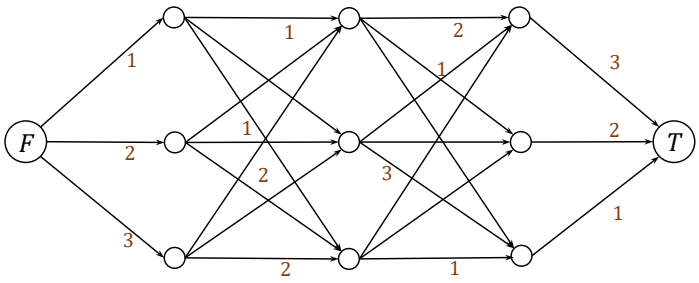

Figure 3: Shortest path problem.

A naive solution to this problem is via enumeration. That is, find all the paths connecting F to T, compute the cost of each path, and select the path with the minimal cost. For a problem with N layer (stage) and m states, there are m^{N-2} possible paths, and on each path, one has to do addition operation for $N-1$ times. That is, one has to do at least $(N-1)m^{N-2}$ addition operations, which grows exponentially fast as the number of stages increases. Even for small m, this is not realistic since in practice, N is usually very large.

Dynamic programming can be seen as an *algorithm* that can reduce the computational loads based on the celebrated *Bellman's principle of optimality:*

Bellman's principle of optimality

An optimal policy has the property that no matter what the previous decisions have been, the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions.

This principle is almost obvious and that no proof is needed, although rigorous proof is not hard to provide. This principle not only holds for discrete time systems, it holds also for continuous systems, stochastic systems, systems described by PDEs and so on.

Now let's apply it to our shortest path problem and see what it gives us. Denote $J_i(x)$ the *cost-to-go* function from state x at stage i to stage N, $\mathcal{N}(x)$ the set of neighbours of x at the next stage and $c(x, y)$ the cost going from state x (at stage i) to y (at stage $i + 1$). The shortest path problem amounts to finding

$$\min_{\text{paths } F \to T} J_1(F).$$

Define the *value function*

$$J_i^*(x) = \min_{\text{paths } x \to T} J_i(x)$$

which is the optimal cost going from x at stage i to T. Suppose that we have found an optimal path ℓ, then at any stage $< N$, for $x \in \ell$, according to Bellman's principle, there must hold

$$J_i^*(x) = \min_{y \in \mathcal{N}(x)} \{c(x, y) + J_{i+1}^*(y)\} \tag{9}$$

for $i = 1, \cdots, N-1$. The boundary condition appears at $i = N$, in which case $J_N^*(y) = 0$. In principle, one may solve the above equation backward to finally get the value $J_1^*(F)$ and the desired shortest path. Let us count the number of additions that we need to do. As before, the digraph has N stages and at each stage, there are m states. Thus to obtain $J_{N-1}^*(\cdot)$, there is nothing to do. To obtain $J_{N-2}^*(\cdot)$, at most m^2 additions and m^2 comparisons are needed, the same for $J_i^*(\cdot)$ when $2 \leq i \leq N-2$. For $J_1^*(\cdot)$, only m additions and m comparisons are needed. Putting together these operations, we need $O(Nm^2)$ additions. This number is much smaller than $(N-1)m^{N-2}$ when N is large. The equation (9), derived from Bellman's principle, is called the *Bellman equation* of this problem.

Although Bellman's equation is merely a necessary condition, it is clear that in the shortest path problem, it's also sufficient for finding the optimal path. We underscore a crucial property of the cost function that can be easily neglected when applying Bellman's principle. That is, the fact that the total cost is a sum of the costs at each step is essential. We will come back to this point when we study continuous dynamic programming. For the moment, establishing some intuitions is enough.

6.2.2 Optimal control on finite horizon

We now dive into optimal control of discrete time systems. We will see that optimal control can be formulated as a shortest path problem, at least when the control space and state space are finite. Thus the above reasoning still holds true.

Consider the nonlinear discrete time dynamical system

$$x_{k+1} = f_k(x_k, u_k), \tag{10}$$

where $x_k \in X_k$ (the system state at time instant k), $u_k \in U_k$ (the input at time instant k). We consider cost of the following form

$$J = \varphi(x_N) + \sum_{k=1}^{N-1} L_k(x_k, u_k), \tag{11}$$

with $1 \leq N \in \mathbb{Z}$, and the initial state x_1 is assumed to fixed. Here φ and L_k are assumed to be some non-negative functions. The control objective is to find a control input sequence $\pi = (u_1, \cdots, u_{N-1})$, which is also called a *policy,* such that the cost J is minimized, while keeping the constraints $x_k \in X_k$ and $u_k \in U_k$.

Notice that the cost (11) is only calculated on finite time intervals, i.e., from 1 to N. We call such an optimal control problem on finite horizon. Later we will also consider infinite horizon cost of the form

$$J = \sum_{k=1}^{\infty} L_k(x_k, u_k). \tag{12}$$

As mentioned before, when U_k and X_k are finite sets, the optimal control problem is equivalent to a shortest path problem. Thus we can immediately derive the Bellman equation. In general, e.g., under

the constraint $|u_k| \leq 1$, the problem is no longer a shortest path problem, but Bellman's principle is still valid. As before, define the cost-to-go function $J_i(x) = \sum_{k=i}^{N-1} L_k(x_k, u_k)|_{x_k=x} + \varphi(x_N)$, and the value function $J_i^*(x) = \min_{(u_i,\cdots,u_{N-1})} J_i(x)$. Then according to Bellman's principle,

$$J_i^*(x) = \min_{u_i \in U_i} \{L_i(x, u_i) + J_{i+1}^*(f_i(x, u_i))\}. \tag{13}$$

The above equation meets the boundary at $i = N-1$, with $J_N^*(x) = \varphi(x)$, for there is no control at the final stage. Equation (13) is the *Bellman equation* for the optimal control problem on finite horizon. The optimal control problem for discrete time systems on finite horizon can be reduced to solving the Bellman equation. It is easy to notice that, this equation can be solved backward. For example, since $J_N^*(\cdot)$ is known, we deduce

$$u_{N-1}^*(x_{N-1}) = \arg\min_{u_{N-1}} \{L_{N-1}(x_{N-1}, u_{N-1}) + \varphi(f_{N-1}(x_{N-1}, u_{N-1}))\}$$

and so on. Finally, one terminates at $u_1^*(x_1) = \arg\min_{u_1}\{L_1(x_1, u_1) + J_2^*(f_1(x_1, u_1))\}$. The function $J_1^*(x_1)$ is clearly the optimal cost and the corresponding policy $(u_1^*(x_1), \cdots, u_{N-1}^*(x_{N-1}))$ is optimal.

6.2.3 Example: Discrete LQR on finite horizon

Consider the constraint free linear plant

$$x_{k+1} = Ax_k + Bu_k$$

with cost function defined by

$$J = x_N^\top S_N x_N + \sum_{i=1}^{N-1} (x_i^\top Q x_i + u_i^\top R u_i)$$

with $Q \geq 0$, $S_N \geq 0$ and $R > 0$. This is called a linear quadratic regulator problem.

The objective is to find an optimal control policy such that J is minimized. Using previous notations, the Bellman equation reads

$$J_i^*(x) = \min_{u_i}\{x^\top Q x + u_i^\top R u_i + J_{i+1}^*(Ax + Bu_i)\} \tag{14}$$

with boundary condition $J_N^*(x) = x^\top S_N x$. We assert that $J_i^*(x)$ is of the form $x^\top S_i x$ for some $S_i \geq 0$. To see this, we calculate $J_{N-1}^*(x_{N-1})$ and the rest is justified by induction. Indeed,

$$J_{N-1}^*(x_{N-1}) = \min_{u_{N-1}} \{x_{N-1}^\top Q x_{N-1} + u_{N-1}^\top R u_{N-1} + (Ax_{N-1} + Bu_{N-1})^\top S_N (Ax_{N-1} + Bu_{N-1})\},$$

from which we see that

$$u_{N-1}^* = -(B^\top S_N B + R)^{-1} B^\top S_N A x_{N-1}$$

and it is evident that $J_{N-1}^*(x_{N-1})$ contains no first order or scalar terms. Define

$$K_{N-1} := (B^\top S_N B + R)^{-1} B^\top S_N A$$

which is called the *Kalman gain*, then $u_{N-1}^* = -K_{N-1} x_{N-1}$. Substituting u_{N-1}^* back, after direct but cumbersome calculations, we get

$$J_{N-1}^* = x_{N-1}^\top S_{N-1} x_{N-1}$$

where

$$S_{N-1} = Q + (A - BK_{N-1})^\top S_N (A - BK_{N-1}) + K_{N-1}^\top R K_{N-1}$$

or equivalently

$$S_{N-1} = Q + A^\top S_N A - A^\top S_N B (R^\top S_N R + B)^{-1} B^\top S_N A.$$

By induction, one may derive the equation for u_i^*, K_i and S_i, which we summarize in the following:

$$\begin{array}{ll} K_i & = (B^\top S_{i+1} B + R)^{-1} B^\top S_{i+1} A \\ u_i^* & = -K_i x_i \\ J_i^* & = x_i^\top S_i x_i \\ S_i & = Q + (A - BK_i)^\top S_{i+1}(A - BK_i) + K_i^\top R K_i \end{array} \tag{15}$$

with boundary condition S_N a known matrix. The optimal value of the problem is provided by $J_1^*(x_1) = x_1^\top S_1 x_1$. The algorithm runs as

$$S_N \to (K_{N-1}, u_{N-1}^*) \to S_{N-1} \to (K_{N-2}, u_{N-2}^*) \to \cdots \to S_2 \to (K_1, u_1^*) \to S_1$$

Although the linear plant we consider here is time-invariant, the controller is time dependent. And the extension to time-varying linear systems is rather straightforward: it suffices to replace A by A_i and B by B_i in the formula (15).

Remark 1. Here we mention a difficulty in solving the Bellman equation. When no additional structures are imposed on f and L, the minimization (13) is often not numerically tractable. When U_i and X_i are finite with low dimension, it is not a big problem. When their dimensions are large, e.g., $U_i = \prod_{k=1}^m I_k \subseteq \mathbb{R}^m$ and $X_i = \prod_{k=1}^n J_k \subseteq \mathbb{R}^n$, where the dimension of I_k and J_k are q, p respectively, then there will be q^m possible inputs and p^n states at each stage. In the worst case, there will be $O(Np^n q^m)$ addition operations to do, which is intractable when p and q are large for $n, m \geq 3$. Such phenomenon is called *curse of dimensionality* noticed by Bellman in the 1960s. Today, this term is widely used in various areas to indicate the intractability of the algorithm in higher dimension. To cope with this, one can resort to approximation schemes.

6.2.4 Infinite horizon problem

Unlike in the finite horizon case, where the time-dependence of the system is of little importance (for example, even though the system is time-invariant, the optimal policy is clearly time-dependent), the optimal control of time-invariant systems on infinite horizon is quite different from that of time-varying systems. In particular, the theory for time-invariant system is much richer than that of time-varying system. For this reason, we will focus on time-invariant system

$$x_{k+1} = f(x_k, u_k) \tag{16}$$

where $x_k \in X$ and $u_k \in U$ for all $k \geq 1$. The admissible control input space may be state-dependent, say $u_k \in U(x_k) \subseteq U$, a constraint. The cost function is of the form

$$J = \sum_{k=1}^{\infty} L(x_k, u_k). \tag{17}$$

Claim. For any stationary policy u, i.e., $u_k = u(x_k)$ for all $k \geq 1$, the cost function (17) under policy u has the property that

$$J_u(x) = L(x, u(x)) + J_u(f(x, u(x)))$$

In fact, $J(x) = L(x, u(x)) + \sum_{k=2}^{\infty} L(x_k, u(x_k)) = L(x, u(x)) + J_u(f(x, u(x))$, as claimed.

Recall that the cost-to-go function $J_i(x) = \sum_{k=i}^{\infty} L(x_k, u_k)|_{x_i = x}$. The value function J_i^* is the same for all i since

$$J_i^*(x) = \min_{(u_i, \cdots,} \sum_{k=i}^{\infty} L(x_k, u_k)|_{x_i = x} = \min_{(u_1, \cdots,} \sum_{k=1}^{\infty} L(x_k, u_k)|_{x_1 = x} = J_1^*(x)$$

Due to this, we may denote $J^*(x) := J_i^*(x)$, and the Bellman equation takes a very special structure:

$$J^*(x) = \min_{u \in U(x)} \{L(x, u) + J^*(f(x, u))\}. \tag{18}$$

The difference of (18) compared to the Bellman equation of finite horizon problem lies in the fact that the function J^* appears on both sides of the equation. Therefore, it seems not possible to solve equation (18) via backward iteration as in the finite horizon case, after all, there is no boundary condition to start with! However, one may guess that starting with $J^* = 0$ and by iteration, J^* converges to a solution. We will discuss this in more detail in next subsection. Once J^* has been found, the optimal policy is given by

$$u^*(x) = \arg \min_{u \in U(x)} \{L(x, u) + J^*(f(x, u))\}.$$

As mentioned before, Bellman equation provides necessary and sufficient condition for finite horizon optimal control problems. One may ask if this still holds for infinite horizon problem, i.e., when (18) is satisfied for some function $\hat{J}$, is $\hat{J}$ the optimal cost function? This is clearly untrue as one may always add a constant to the solution which produces another solution. But at least we know the following.

Proposition 1. *Let J^* be the optimal cost function and $\hat{J}$ a solution to the Bellman equation (18), then $\hat{J} \geq J^*$.*

Proof. By assumption, there exists $\hat{u}(\cdot)$ satisfying $\hat{J}(x) = L(x, \hat{u}(x)) + \hat{J}(f(x, \hat{u}(x))$. Then under the policy $\hat{u}(\cdot)$, for any $x_1 \in X$, we have

$$\hat{J}(x_1) = \hat{J}(x_k) + \sum_{i=1}^{k} L(x_i, \hat{u}(x_i)),$$

which holds for all $k \geq 1$. Thus $\hat{J}(x_1) \geq \sum_{i=1}^{\infty} L(x_i, \hat{u}(x_i)) \geq J^*(x_1)$. □

On the other hand, if we know before hand that the solution to the Bellman equation is unique (at least in a certain class), then we may conclude that solving Bellman equation is sufficient to find the optimal cost function.

6.2.5 Solving by iteration[1]

In general, we have to solve the Bellman equation numerically. There are two basic iteration approaches which solves (18) approximately, namely, policy iteration and value iteration.

Value iteration: start from some non-negative function $J_0 : X \to \mathbb{R}$ and iterate according to

$$J_{k+1}(x) = \min_{u \in U(x)} \{L(x, u) + J_k(f(x, u))\}. \tag{19}$$

The approximated optimal policy can be taken as

$$u^*_{N+1}(x) = \arg \min_{u \in U(x)} \{L(x, u) + J_N(f(x, u))\}$$

when J_N reaches a reasonable level of accuracy.

There is an important property of value iteration, called the *monotonicity* property. Being J^* the optimal cost function, if we start from $J_0 \geq J^*$, then $J_k \geq J^*$ for all $k \geq 0$. In fact,

$$\begin{aligned} J_1(x) &= \min_{u \in U(x)} \{L(x, u) + J_0(f(x, u))\} \\ &\geq \min_{u \in U(x)} \{L(x, u) + J^*(f(x, u))\} \\ &= J^*(x). \end{aligned}$$

Interestingly, we can get stronger result for the case $J_0 = 0$. That is, the sequence $\{J_k\}$ is monotone increasing:

$$0 \leq J_1 \leq J_2 \leq \cdots \leq J_*$$

since

$$\begin{aligned} J_1(x) &= \min_{u \in U(x)} L(x, u) \geq 0 \\ J_2(x) &= \min_{u \in U(x)} \{L(x, u) + J_1(f(x, u))\} \\ &\geq \min_{u \in U(x)} L(x, u) \\ &= J_1(x) \\ &\vdots \end{aligned}$$

Thus, there exists a function $\tilde{J} \leq J^*$, such that $J_k \to \tilde{J}$ pointwisely, but there may exist a gap between $\tilde{J}$ and J^*. The following classical result provides a sufficient condition that $\tilde{J} = J^*$.

[1]This part is mainly taken from the paper [1].

Proposition 2 (Convergence of value iteration I)**.** *If U is a metric space and the sets*

$$U_k(x,\lambda) = \{u \in U(x) : L(x,u) + J_k(f(x,u)) \le \lambda\}$$

is compact for all $x \in X$, $\lambda \in \mathbb{R}$ and k, then the value iteration $J_k \uparrow J^*$ pointwisely for any $J_0 \ge 0$ satisfying $J_0(x) \le \min_{u\in U(x)} L(x,u) + J_0(f(x,u))$ for all $x \in X$, e.g., $J_0 = 0$.

Again, we use the LQR problem as an example.

Consider the linear time-invariant discrete time system

$$x_{k+1} = Ax_k + Bu_k \tag{20}$$

with quadratic cost

$$J = \sum_{k=1}^{\infty} (x_k^T Q x_k + u_k^T R u_k) \tag{21}$$

where $Q \ge 0$ and $R > 0$. Assume that $u \in \mathbb{R}^m$ is constraint free. In order that $J < \infty$, it is sufficient to assume that the system is stabilizable (verify!). Now the Bellman equation (18) reads

$$J^*(x) = \min_u \{x^\top Q x + u^\top R u + J^*(Ax + Bu)\}. \tag{22}$$

Let's start with $J_1 = 0$. By definition of VI,

$$\begin{aligned} J_1(x) &= \min_u \{x^\top Q x + u^\top R u + 0\} = x^\top Q x \\ u_1(x) &\in \arg\min_u \{x^\top Q x + u^\top R u + 0\} = 0. \end{aligned}$$

Denote $J_1(x)$ as $J_1(x) =: x^\top P_1 x$, or $P_1 = Q$. To get $J_2(x)$ and $u_2(x)$, we calculate

$$\begin{aligned} J_2(x) &= \min_u \{x^\top Q x + u^\top R u + J_1(x)\} \\ &= \min_u \{x^\top Q x + u^\top R u + (Ax + Bu)^\top P_1 (Ax + Bu)\} \\ &= x^\top P_2 x \\ u_2(x) &\in \arg\min_u \{x^\top Q x + u^\top R u + J_1(x)\} = -K_1 x. \end{aligned}$$

Notice that this is the "reverse computation" of the optimal controller for finite horizon LQR.

References

[1] Dimitri P Bertsekas. Value and policy iterations in optimal control and adaptive dynamic programming. *IEEE transactions on neural networks and learning systems*, 28(3):500–509, 2015.