

Dynamic Programming

Moritz Diehl

Overview

- ▶ Discrete Time Systems
 - ▶ DP recursion
 - ▶ Linear Quadratic Regulator
 - ▶ Bellman equation
 - ▶ Robust and Stochastic Dynamic Programming
 - ▶ Convex Dynamic Programming
- ▶ Continuous Time Systems
 - ▶ Hamilton-Jacobi-Bellman equation
 - ▶ Linear Quadratic Regulator
 - ▶ stationary Hamilton-Jacobi-Bellman equation

Discrete Time Optimal Control Problem

$$\underset{s, q}{\text{minimize}} \quad \sum_{i=0}^{N-1} l_i(s_i, q_i) + E(s_N)$$

subject to

$$\begin{aligned} s_0 - x_0 &= 0, && \text{(initial value)} \\ s_{i+1} - f_i(s_i, q_i) &= 0, & i = 0, \dots, N-1, & \text{(discrete system)} \\ h_i(s_i, q_i) &\geq 0, & i = 0, \dots, N, & \text{(path constraints)} \\ r(s_N) &\geq 0. && \text{(terminal constraints)} \end{aligned}$$

In DP, can easily get rid of inequality constraints h_i, r by giving infinite costs $l_i(s, u)$ or $E(s)$ to infeasible points (s, u) .

Dynamic Programming Recursion

Iterate backwards, for $k = N - 1, N - 2, \dots$

$$J_k(x) = \min_u \underbrace{l_k(x, u) + J_{k+1}(f_k(x, u))}_{=\tilde{J}_k(x, u)}$$

starting with

$$J_N(x) = E(x)$$

In recursion with constraints, make sure you assign infinite costs to all infeasible points, in all three functions l_k , E , and J .

How to get optimal controls?

Based on \tilde{J}_k , obtain feedback controls laws for $k = 0, 1, \dots, N - 1$

$$u_k^*(x) = \arg \min_u \underbrace{l_k(x, u) + J_{k+1}(f(x, u))}_{=\tilde{J}_k(x, u)}.$$

For given initial value x_0 , we can thus obtain the optimal trajectories of x_k and u_k by the closed-loop system:

$$x_{k+1} = f_k(x_k, u_k^*(x_k))$$

This is a forward recursion yielding in particular u_0, \dots, u_{N-1} .

Linear Quadratic Problems

Regard now linear quadratic optimal control problem of the form

$$\underset{x,u}{\text{minimize}} \quad \sum_{i=0}^{N-1} \begin{bmatrix} x_i \\ u_i \end{bmatrix}^T \begin{bmatrix} Q_i & S_i^T \\ S_i & R_i \end{bmatrix} \begin{bmatrix} x_i \\ u_i \end{bmatrix} + x_N^T P_N x_N$$

subject to

$$x_0 - x_0^{\text{fix}} = 0, \quad (\text{initial value})$$

$$x_{i+1} - A_i x_i - B_i u_i = 0, \quad i = 0, \dots, N-1, \quad (\text{discrete system})$$

How to apply dynamic programming here?

Linear Quadratic Recursion

If

$$l_i(x, u) = \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} Q & S^T \\ S & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

and

$$J_{k+1} = x^T P x$$

and

$$f_i(x, u) = Ax + Bu$$

then

$$J_k(x) = \min_u \begin{bmatrix} x \\ u \end{bmatrix}^T \left(\begin{bmatrix} Q & S^T \\ S & R \end{bmatrix} + \begin{bmatrix} A^T P A & A^T P B \\ B^T P A & B^T P B \end{bmatrix} \right) \begin{bmatrix} x \\ u \end{bmatrix}$$

This has an easy explicit solution if $R + B^T P B$ is invertible ... we need the Schur Complement Lemma.

Schur Complement Lemma

Let us simplify notation and regard

$$\phi(x) = \min_u \underbrace{\begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} Q & S^T \\ S & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}}_{=\psi(x,u)}$$

with R invertible. Then

$$\phi(x) = x^T \left(Q - S^T R^{-1} S \right) x$$

and

$$\arg \min_u \psi(x, u) = -R^{-1} S x$$

PROOF: exercise.

Riccati Recursion

The Schur Komplement Lemma applied to the LQ recursion:

$$J_k(x) = \min_u \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} Q + A^T P A & S^T + A^T P B \\ S + B^T P A & R + B^T P B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

delivers directly, if $R + B^T P B$ is invertible:

$$J_k(x) = x^T P_{\text{new}} x$$

with

$$P_{\text{new}} = Q + A^T P A - (S^T + A^T P B)(R + B^T P B)^{-1}(S + B^T P A)$$

Thus, if J_{k+1} was quadratic, also J_k is!

Difference Riccati Equation

Backwards recursion: starting with P_N , we iterate for $k = N - 1, \dots, 0$

$$P_k = Q_k + A_k^T P_{k+1} A_k - (S_k^T + A_k^T P_{k+1} B_k)(R_k + B_k^T P_{k+1} B_k)^{-1}(S_k + B_k^T P_{k+1} A_k)$$

Then, we obtain the optimal feedback u_k^* by

$$u_k^* = -(R_k + B_k^T P_{k+1} B_k)^{-1}(S_k + B_k^T P_{k+1} A_k)x_k$$

and forward recursion

$$x_{k+1} = A_k x_k + B_k u_k^*.$$

Inhomogenous Linear Systems

Can we extend the Riccati recursion also to inhomogenous costs and systems? I.e. problems of the form:

minimize
 x, u

$$\sum_{i=0}^{N-1} \begin{bmatrix} 1 \\ x_i \\ u_i \end{bmatrix}^T \begin{bmatrix} 0 & q_i^T & s_i^T \\ q_i & Q_i & S_i^T \\ s_i & S_i & R_i \end{bmatrix} \begin{bmatrix} 1 \\ x_i \\ u_i \end{bmatrix} + \begin{bmatrix} 1 \\ x_N \end{bmatrix}^T \begin{bmatrix} 0 & p_N^T \\ p_N & P_N \end{bmatrix} \begin{bmatrix} 1 \\ x_N \end{bmatrix}$$

subject to

$$\begin{aligned} x_0 - x_0^{\text{fix}} &= 0, && \text{(initial value)} \\ x_{i+1} - A_i x_i - B_i u_i - c_i &= 0, && i = 0, \dots, N-1, \text{ (discrete system)} \end{aligned}$$

Why Inhomogenous Systems and Costs?

They appear in

- ▶ Linearization of Nonlinear Systems
- ▶ Reference Tracking Problems e.g. with $l_i(x_i, u_i) = \|x_i - x_i^{\text{ref}}\|_Q^2 + \|u_i\|_R^2$
- ▶ Filtering Problems (Moving Horizon Estimation, Kalman Filter) with cost $l_i(x_i, u_i) = \|Cx_i - y_i^{\text{meas}}\|_Q^2 + \|u_i\|_R^2$
- ▶ Subproblems in Active Set Methods for Constrained LQ

A Simple Programming Trick

By augmenting the system states x_k to

$$\tilde{x}_k = \begin{bmatrix} 1 \\ x_k \end{bmatrix}$$

and replacing the dynamics by

$$\tilde{x}_{k+1} = \begin{bmatrix} 1 & 0 \\ c_k & A_k \end{bmatrix} \tilde{x}_k + \begin{bmatrix} 0 \\ B_k \end{bmatrix} u_k$$

with initial value

$$\tilde{x}_0^{\text{fix}} = \begin{bmatrix} 1 \\ x_0^{\text{fix}} \end{bmatrix}$$

This is a homogenous problem and can be solved exactly as before!

Linear Quadratic Regulator (LQR)

Regard now LQ problem with infinite horizon and autonomous (time independent) system and cost:

$$\underset{x,u}{\text{minimize}} \quad \sum_{i=0}^{\infty} \begin{bmatrix} x_i \\ u_i \end{bmatrix}^T \begin{bmatrix} Q & S^T \\ S & R \end{bmatrix} \begin{bmatrix} x_i \\ u_i \end{bmatrix}$$

subject to

$$x_0 - x_0^{\text{fix}} = 0, \quad (\text{initial value})$$

$$x_{i+1} - Ax_i + Bu_i = 0, \quad i = 0, \dots, \infty, \quad (\text{discrete system})$$

How to apply dynamic programming here?

Algebraic Riccati Equation

Require stationary solution of Riccati Recursion:

$$P_k = P_{k+1}$$

i.e.

$$P = Q + A^T P A - (S^T + A^T P B)(R + B^T P B)^{-1}(S + B^T P A)$$

This is called the Algebraic Riccati Equation (in discrete time)

Then, we obtain the optimal feedback $u^*(x)$ by

$$u^*(x) = - \underbrace{(R + B^T P B)^{-1}(S + B^T P A)}_{=K} x$$

This feedback is called the Linear Quadratic Regulator (LQR), and K is the LQR gain.

Infinite Horizon Problem

Can regard more general infinite horizon problem:

$$\underset{s, q}{\text{minimize}} \quad \sum_{i=0}^{\infty} l(s_i, q_i)$$

subject to

$$\begin{aligned} s_0 - x_0 &= 0, && \text{(initial value)} \\ s_{i+1} - f(s_i, q_i) &= 0, \quad i = 0, \dots, \infty, && \text{(discrete system)} \end{aligned}$$

The Bellman Equation

Requiring stationarity of solutions of Dynamic Programming Recursion:

$$J_k = J_{k+1}$$

leads directly to the famous Bellman Equation:

$$J(x) = \min_u \underbrace{l(x, u) + J(f(x, u))}_{=\tilde{J}(x, u)}$$

The optimal controls are then obtained by the function

$$u^*(x) = \arg \min_u \tilde{J}(x, u).$$

This feedback is called the stationary “Optimal Feedback Control”. It is a static state feedback law that generalizes LQR. But in contrast to LQR it is generally nonlinear.

Overview

- ▶ Discrete Time Systems
 - ▶ DP recursion
 - ▶ Linear Quadratic Regulator
 - ▶ Bellman equation
 - ▶ **Robust and Stochastic Dynamic Programming**
 - ▶ Convex Dynamic Programming
- ▶ Continuous Time Systems
 - ▶ Hamilton-Jacobi-Bellman equation
 - ▶ Linear Quadratic Regulator
 - ▶ stationary Hamilton-Jacobi-Bellman equation

Robust Dynamic Programming

Dynamic Programming can easily be applied to games (like chess). Here, an adverse player chooses disturbances w_k against us. They influence both the stage costs l_i as well as the system dynamics f_i . The robust DP recursion is simply:

$$J_k(x) = \min_u \max_w \underbrace{l_k(x, u, w) + J_{k+1}(f_k(x, u, w))}_{\tilde{J}_k(x, u)}$$

starting with

$$J_N(x) = E(x)$$

Stochastic Dynamic Programming

Also, we might find the feedback law that gives us the best expected value. Here, we take an expectation over the disturbances w_k .

The stochastic DP recursion is simply:

$$J_k(x) = \min_u \underbrace{\mathbb{E}_w \{ l_k(x, u, w) + J_{k+1}(f_k(x, u, w)) \}}_{\tilde{J}_k(x, u)}$$

where $\mathbb{E}_w \{ \cdot \}$ is the expectation operator, i.e. the integral over w weighted with the probability density function $p(w|x, u)$ of w given x and u :

$$\mathbb{E}_w \{ \phi(x, u, w) \} = \int \phi(x, u, w) P(w|x, u) dw$$

In case of finitely many scenarios, this is just a weighted sum. Dynamic Programming can avoid the combinatorial explosion of scenario trees.

Monotonicity of Dynamic Programming

The “cost-to-go” J_k is often also called the “value function”.
The “dynamic programming operator” T_k acting on one value function and giving another one is defined by

$$T_k(J)[x] = \min_u l_k(x, u) + J(f_k(x, u)).$$

(the dynamic programming recursion is then compactly written as $J_k = T_k(J_{k+1})$)
If $J \geq J'$ (in the sense $J(x) \geq J'(x)$ for all x) then also

$$T_k(J) \geq T_k(J').$$

This is called “monotonicity” of dynamic programming. It holds also for robust or stochastic dynamic programming. It can e.g. be used in existence proofs for solutions of the stationary Bellman equation.

Convex Dynamic Programming

An interesting observation is that certain DP operators T_k preserve convexity of the value function J .

THEOREM

- ▶ If system is linear, $f(x, u, w) = A(w)x + B(w)u + c(w)$,
- ▶ stage cost $l(x, u, w)$ is convex in (x, u)

then DP, robust DP, and stochastic DP operators T preserve convexity of J .

This means: if J is a convex function, then $T(J)$ is again a convex function.

Proof of Convexity Preservation

Regard $I(x, u, w) + J(f(x, u, w))$.

For fixed w , this is a convex function in (x, u) . Because also maximum over w or expectation preserve convexity, the function

$$\tilde{J}(x, u)$$

is in all three cases convex in both x and u .

Finally, the minimization of a convex function over one of its arguments preserves convexity, i.e. the resulting value function $T(J)$ defined by

$$T(J)[x] = \min_u \tilde{J}(x, u)$$

is convex. □

Why is convexity important?

- ▶ computation of feedback law $\arg \min_u \tilde{J}(x, u)$ is a convex parametric program: can be solved by local optimization methods.
- ▶ Can represent value function $J(x)$ more efficiently than by tabulation, e.g. as maximum of linear functions

$$J(x) = \max_i a_i^T \begin{bmatrix} 1 \\ x \end{bmatrix}$$

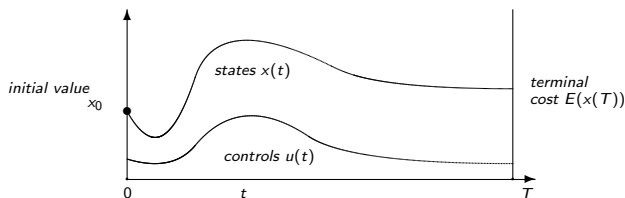
- ▶ In robust DP, convexity of value function allows to conclude that worst case is assumed on boundary of uncertainty sets.

Overview

- ▶ Discrete Time Systems
 - ▶ DP recursion
 - ▶ Linear Quadratic Regulator
 - ▶ Bellman equation
 - ▶ Robust and Stochastic Dynamic Programming
 - ▶ Convex Dynamic Programming
- ▶ **Continuous Time Systems**
 - ▶ Hamilton-Jacobi-Bellman equation
 - ▶ Linear Quadratic Regulator
 - ▶ stationary Hamilton-Jacobi-Bellman equation

Continuous Time Optimal Control

Regard simplified optimal control problem:



$$\text{minimize}_{x(\cdot), u(\cdot)} \int_0^T L(x(t), u(t)) dt + E(x(T))$$

subject to

$$\begin{aligned} x(0) - x_0 &= 0, && \text{(fixed initial value)} \\ \dot{x}(t) - f(x(t), u(t)) &= 0, && t \in [0, T]. \quad \text{(ODE model)} \end{aligned}$$

Euler Discretization

Introduce timestep

$$h = \frac{T}{N}$$

$$\underset{s, q}{\text{minimize}} \quad \sum_{i=0}^{N-1} hL(s_i, q_i) + E(s_N)$$

subject to

$$\begin{aligned} s_0 - x_0 &= 0, && \text{(initial value)} \\ s_{i+1} - s_i - hf(s_i, q_i) &= 0, \quad i = 0, \dots, N-1, && \text{(discretized system)} \end{aligned}$$

Dynamic Programming for Euler Scheme

Using DP for Euler Discretized OCP yields:

$$J_k(x) = \min_u hL(x, u) + J_{k+1}(x + hf(x, u))$$

Replacing the index k by time points $t_k = kh$ we obtain

$$J(x, t_k) = \min_u hL(x, u) + J(x + hf(x, u), t_k + h).$$

Assuming differentiability of $J(x, t)$ in (x, t) and Taylor expansion yields

$$J(x, t) = \min_u hL(x, u) + J(x, t) + h\nabla J(x, t)^T f(x, u) + h\frac{\partial J}{\partial t}(x, t) + O(h^2)$$

Hamilton-Jacobi-Bellman (HJB) Equation

Bringing all terms independent of u to the left side and dividing by $h \rightarrow 0$ yields

$$-\frac{\partial J}{\partial t}(x, t) = \min_u L(x, u) + \nabla J(x, t)^T f(x, u)$$

This is the famous Hamilton-Jacobi-Bellman equation.

We solve this partial differential equation (PDE) backwards for $t \in [0, T]$, starting at the end of the horizon with

$$J(x, T) = E(x).$$

NOTE: Optimal feedback control for state x at time t is obtained from

$$u^*(x, t) = \arg \min_u L(x, u) + \nabla J(x, t)^T f(x, u)$$

Hamiltonian function

Introducing the Hamiltonian function

$$H(x, \lambda, u) := L(x, u) + \lambda^T f(x, u)$$

and the so called “true” Hamiltonian

$$H^*(x, \lambda) := \underbrace{\min_u H(x, \lambda, u)}_{=H(x, \lambda, u^*(x, u))}$$

we can write the Hamilton-Jacobi-Bellman equation compactly as:

$$-\frac{\partial J}{\partial t}(x, t) = H^*(x, \nabla J(x, t))$$

Linear Quadratic Problems

Regard now linear quadratic optimal control problem of the form

$$\underset{x(\cdot), u(\cdot)}{\text{minimize}} \quad \int_0^T \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} Q(t) & S(t)^T \\ S(t) & R(t) \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} dt + x(T)^T P_T x(T)$$

subject to

$$\begin{aligned} x(0) - x_0 &= 0, && \text{(fixed initial value)} \\ \dot{x} - A(t)x - B(t)u &= 0, \quad t \in [0, T]. && \text{(linear ODE model)} \end{aligned}$$

Linear Quadratic HJB

Assuming that $J(x, t) = x^T P(t)x$, the HJB Equation reads as:

$$-\frac{\partial J}{\partial t}(x, t) = \min_u \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} Q(t) & S(t)^T \\ S(t) & R(t) \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + 2x^T P(t)(A(t)x + B(t)u)$$

Symmetrizing, the right hand side is:

$$\min_u \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} Q + PA + A^T P & S^T + PB \\ S + B^T P & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

which by the Schur Complement Lemma yields

$$-\frac{\partial J}{\partial t} = x^T \left(Q + PA + A^T P - (S^T + PB)R^{-1}(S + B^T P) \right) x$$

Thus, if J was quadratic, it remains quadratic!

Differential Riccati Equation

The matrix differential equation

$$-\dot{P} = Q + PA + A^T P - (S^T + PB)R^{-1}(S + B^T P)$$

with terminal condition

$$P(T) = P_T$$

is called the “differential Riccati equation”.

Its feedback law is by the Schur complement lemma:

$$u^*(x, t) = -R(t)^{-1}(S(t) + B(t)^T P(t))x$$

Linear Quadratic Regulator

The solution to the infinite horizon problem with time independent costs and system matrices is given by setting

$$\dot{P} = 0$$

and solving

$$0 = Q + PA + A^T P - (S^T + PB)R^{-1}(S + B^T P).$$

This equation is called the algebraic Riccati equation (in continuous time). Its feedback law is again a static linear gain:

$$u^*(x) = - \underbrace{R^{-1}(S + B^T P)}_{=K} x$$

Continuous vs. Discrete Time

Regard Euler transition from continuous time to discrete time:

$$f(x, u) \rightarrow x + hf(x, u)$$

and

$$L(x, u) \rightarrow hL(x, u)$$

In both cases, J and u^* will remain the same:

$$J(x, t_k) \rightarrow J_k(x)$$

and

$$u^*(x, t_k) \rightarrow u_k^*(x)$$

NOTE: In LQR, both discrete and continuous time formulation yield same matrices P and K .

Infinite Time Optimal Control

Regard infinite time optimal control problem:

$$\underset{x(\cdot), u(\cdot)}{\text{minimize}} \quad \int_0^{\infty} L(x(t), u(t)) dt + E(x(T))$$

subject to

$$\begin{aligned} x(0) - x_0 &= 0, && \text{(fixed initial value)} \\ \dot{x}(t) - f(x(t), u(t)) &= 0, \quad t \in [0, \infty]. && \text{(ODE model)} \end{aligned}$$

This leads to stationary HJB equation

$$0 = \min_u L(x, u) + \nabla J(x)^T f(x, u)$$

with stationary optimal feedback control law $u^*(x)$.

Summary

- ▶ Dynamic Programming:

$$J_k(x) = \min_u l_k(x, u) + J_{k+1}(f_k(x, u))$$

- ▶ Hamilton Jacobi Bellman Equation:

$$-\frac{\partial J}{\partial t}(x, t) = \min_u H(x, \nabla J(x, t), u)$$

- ▶ with Hamiltonian function $H(x, \lambda, u) := L(x, u) + \lambda^T f(x, u)$
- ▶ Linear quadratic systems can be analytically solved (LQR), easiest example of larger class of convex dynamic programming
- ▶ DP framework generalizes easily to minimax games and stochastic optimal control.

Literature

- ▶ Dimitri P. Bertsekas: Dynamic Programming and Optimal Control. Athena Scientific, Belmont, 2000 (Vol I, ISBN: 1-886529-09-4) & 2001 (Vol II, ISBN: 1-886529-27-2)
- ▶ A. E. Bryson and Y. C. Ho: Applied Optimal Control, Hemisphere/Wiley, 1975.