ECE7850: Hybrid Systems: Theory and Applications

Lecture Note 8: Discrete Time Optimal Control and Dynamic Programming

Wei Zhang

Assistant Professor
Department of Electrical and Computer Engineering
Ohio State University, Columbus, Ohio, USA

Spring 2017
Outline

- Discrete Time Optimal Control Problem
- Short Introduction to Dynamic Programming
- Connection to Stabilization Problems
Discrete Time Nonlinear System Model I

• Consider a general discrete-time nonlinear system:

\[ x(t + 1) = f(x(t), u(t)), \quad x \in X, u \in U, t \in \mathbb{Z}_+ \]  \hspace{1cm} (1)

• For traditional system: \( x \in X \subseteq \mathbb{R}^n, u \in U \subseteq \mathbb{R}^m \) are continuous variables

• Actual control constraint is state-dependent in general:

\[ U(x) = \{ u \in U : f(x, u) \in X \} \]

• A large class of DT hybrid systems can also be written in the above form:
  - switched systems: \( U \subseteq \mathbb{R}^m \times Q \) with mixed continuous/discrete control input
Discrete Time Nonlinear System Model II

- variable structure system (e.g. Piecewise affine system): \( f \) defined in terms of partitions: 
  \[ f(x(t), u(t)) = f_i(x(t), u(t)), \text{ if } x(t) \in \Omega_i \]

- more general hybrid system: \( X \subseteq \mathbb{R}^n \times Q_x \) and \( U \subseteq \mathbb{R}^m \times Q_u \)

- Discrete time hybrid systems are much easier to deal with. They differ from traditional nonlinear systems mainly in the discrete nature of some part of the state variables or control inputs.

- For simplicity, our discussion on discrete-time optimal control is in the context of traditional nonlinear systems (without worrying about any discrete nature of state or control). However, most concepts and results work directly for DT hybrid systems. We will formally discuss DT hybrid system models later.
Control Law and Policy I

• For system (1), a general way to determine control:
  - time-varying state-feedback control law: \( u(t) = \mu_t(x(t)) \)
  
  - needs to respect (possibly state-dependent) constraints: \( \mu_t(x(t)) \in U(x(t)) \)

• Control policy vs. control inputs:
  - Control policy: a sequence of control laws \( \pi = \{\mu_0(\cdot), \mu_1(\cdot), \ldots\} \)
  
  - Control inputs: a sequence of control input actions \( \{u_0, u_1, \ldots\} \)

• Some terminologies and notations:
  - \( N \)-horizon policy: \( \pi_N = \{\mu_0, \ldots, \mu_{N-1}\} \)
  
  - Infinite-horizon policy: \( \pi_\infty = \{\mu_0, \mu_1, \ldots\} \)
  
  - Stationary policy: \( \pi_\infty = \{\mu, \mu, \ldots\} \)
• $x(\cdot; z, \pi)$: Cl-trajectory starting from $z$ under policy $\pi = \{\mu_0, \mu_1, \ldots\}$

$$x(t + 1; z, \pi) = f(x(t; z, \pi), \mu_t(x(t; z, \pi)))$$  \hspace{1cm} (2)

• Corresponding control sequence: $u(\cdot; z, \pi) = \mu_t(x(t; z, \pi))$

• A policy $\pi = \{\mu_0, \mu_1, \ldots\}$ is called feasible if $x(t; z, \pi) \in X$ and $u(t; z, \pi) \in U(x(t))$

• $\Pi_N$: the set of all feasible $N$-horizon

• $\Pi_\infty$: the set of all feasible infinite-horizon policies
Control Law and Policy III

- Quantify performance of control policy through cost functions:
  
  - Running cost (stage cost) function: \( l : X \times U \rightarrow \mathbb{R}_+ \)
  
  - Terminal cost function: \( J_f : X \rightarrow \mathbb{R}_+ \)

- Finite Horizon: \( J_N(z, \pi_N) = \sum_{t=0}^{N-1} l(x(t; z, \pi_N), u(t; z, \pi_N)) + J_f(x(N; z, \pi_N)) \)

- Infinite Horizon: \( J_\infty(z, \pi_\infty) = \sum_{t=0}^{\infty} l(x(t; z, \pi_\infty), u(t; z, \pi_\infty)) \)
Discrete-Time Optimal Control Problems

- Finite-Horizon Optimal Control Problem:
  \[ V_N(z) = \inf_{\pi_N \in \Pi_N} J_N(z, \pi_N) \]

- Infinite-Horizon Optimal Control Problem:
  \[ V^*(z) = \inf_{\pi_\infty \in \Pi_\infty} J_\infty(z, \pi_\infty) \]

- \( V_N \): \( N \)-horizon value function

- \( V^* \): infinite horizon value function

- \( \lim_{N \to \infty} V_N \) may not exist

- Even when \( V_\infty \) exists, \( V_\infty \neq V^* \) in general
Outline

• Discrete Time Optimal Control Problem

• Short Introduction to Dynamic Programming

• Connection to Stabilization Problems
Dynamic Programming

• Most important tool for solving deterministic and stochastic optimal control problems

• Divide & conquer: The $N$-horizon optimal solution depends on the $N - 1$ horizon optimal solution, which in turns depend on the $N - 2$ horizon optimal solution ...

• So we solve 0-horizon first, then 1-horizon, ..., eventual solve the $N$-horizon optimal control problem.

• The divide & conquer approach is grounded by a fundamental principle: Bellman’s principle of optimality

   Any segment along an optimal trajectory is also optimal among all the trajectories joining the two end points of the segment
Dynamic Programming: Value Iteration I

• For arbitrary integer \( j \geq 0 \), the \( j \)-horizon optimal control problem:

\[
V_j(z) = \min_{u_0, \ldots, u_{j-1}} \{ J_f(x_j) + \sum_{k=0}^{j-1} l(x_k, u_k) \}
\]

subject to \( x_{k+1} = f(x_k, u_k), x_0 = z \)
\( u_k \in U(x_k), k = 0, \ldots, j - 1 \)

(3)

• \( V_j^*(z) \): \( j \)-horizon value function, i.e., the minimum achievable cost if system starts from state \( z \) when there are \( j \) steps left to reach the final time

• Let \( u_0^*, u_1^*, \ldots, u_{j-1}^* \) be the optimal solution to problem (3). If the system is at state \( z \) when there are \( j \) steps left to reach the final time, the first step of the optimal control action is \( u_0^* \), the second step is \( u_1^* \), ....
Dynamic Programming: Value Iteration II

Value Iteration: Compute $V_N(z)$ iteratively from $V_0(z)$

- 0-horizon problem (degenerate case):

- 1-horizon problem:

- 2-horizon problem:
Dynamic Programming: Value Iteration III

Now suppose we are given $V_j(z)$, need to derive $V_{j+1}(z)$

What is the optimal control for $j + 1$ horizon?

- Suppose available controls at time 0 are $U(z) = \{u^{(1)}, u^{(2)}\}$
- Need to compare: $l(z, u^{(1)}) + V_j(f(z, u^{(1)}))$ and $l(z, u^{(2)}) + V_j(f(z, u^{(2)}))$
- The optimal control: $\mu_{j+1}^*(z) = \text{argmin}_{u \in U(z)} \{l(z, u) + V_j(f(z, u))\}$
- The minimum cost: $V_{j+1}(z) = \min_{u \in U(z)} \{l(z, u) + V_j(f(z, u))\}$
Value Iteration Algorithm

• System dynamics: \( x_{k+1} = f(x_k, u_k) \) with \( u_k \in U(x_k) \)

• Determine optimal control law by solving the following optimization problem:

\[
\begin{align*}
\min_{u} & \quad J_N(z, u) = J_f(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k) \\
\text{subj. to} & \quad x_{k+1} = f(x_k, u_k), x_0 = z \quad \text{(system dynamics)} \\
& \quad u_k \in U(x_k) \quad \text{(control constraint)}
\end{align*}
\]

• Solve problem through value iteration: (namely, iteratively compute the value function for 0-horizon, 1-horizon, ..., \( N \)-horizon problems)

  - **Step 0**: (0-horizon): \( V_0(z) = g(z) \)
  
  - **Step j**: given \( V_j(z) \) and the optimal control laws \( \mu_j^*(z), \mu_{j-1}^*(z), \ldots, \mu_1^*(z) \), for the remaining \( j \) steps, compute:
    
    • \( V_{j+1}(z) = \min_{u \in U(z)} \{ l(z, u) + V_j(f(z, u)) \} \)
    
    • \( \mu_{j+1}^*(z) = \arg \min_{u \in U(z)} \{ l(z, u) + V_j(f(z, u)) \} \)

  - \( j \leftarrow j + 1 \), until \( j = N \)
Value Iteration Algorithm

- Value iteration algorithm output:

  - Value functions: \( V_0(z),...,V_N(z) \)

  - Optimal control laws: \( \mu_{j+1}^*(z) = \text{argmin}_{u \in U(z)} \{ l(z, u) + V_j(f(z, u)) \}, \)

  \( j = 1, ..., N - 1 \)

- \( \mu_{j+1}^*(z) \): has the following two meanings

  - the first optimal control action for a \( j + 1 \) horizon problem with initial state \( z \)

  - the optimal control action when the system is at state \( z \) and there are \( j + 1 \) steps to go
Value Iteration Algorithm

- How to use these control laws?

  - Time 0: $x_0 = \hat{x} \rightarrow$ control action: $u_0^* = \mu_N^*(\hat{x})$

  - Time 1: $x_1^* = f(\hat{x}, u_0^*) \rightarrow$ control action: $u_1^* = \mu_{N-1}^*(x_1^*)$

  - Time 2: $x_2^* = f(x_1^*, u_1^*) \rightarrow$ control action: $u_2^* = \mu_{N-2}^*(x_2^*)$

  - ...$

  - Time $N - 1$: $x_{N-1}^* = f(x_{N-2}^*, u_{N-2}^*) \rightarrow$ control action: $u_{N-1}^* = \mu_1^*(x_{N-1}^*)$

  - Time $N$: $x_N^* = f(x_{N-1}^*, u_{N-1}^*)$

- In general: at time $k$: optimal control $u_k^* = \mu_{N-k}^*(x_k^*)$
Properties of Value Iteration Operator I

• To simplify notations, we define two function spaces:

\[ \mathcal{G} \triangleq \{ g : X \to \mathbb{R} \cup \{ \pm \infty \} \} \text{ and } \mathcal{G}_+ \triangleq \{ g : X \to \mathbb{R}_+ \cup \{ +\infty \} \} \]

• Value iteration can be viewed as an operator \( \mathcal{T} : \mathcal{G} \to \mathcal{G} \)

\[
\mathcal{T}[g](z) = \min_{u \in U(z)} l(z, u) + g(f(z, u)), \forall z \in X, g \in \mathcal{G}
\] (4)

• Iteration under a given control law \( \mu : X \to U \) with \( \mu(x) \in X, \forall x \in X \) is defined as

\[
\mathcal{T}_\mu[g](z) = l(z, \mu(z)) + g(f(z, \mu(z))), \forall z \in X, g \in \mathcal{G}
\] (5)
Properties of Value Iteration Operator II

- $T^k$: the composition of $T$ with itself $k$ times:

$$T^{k+1}[g] = T[T^k[g]], \quad \text{with } T^0[g] = g$$

---

**Theorem 1 (Key Properties of Value Iteration [Ber12]).**

Assume that $J_f(z) \geq 0$ and $l(z,u) \geq 0$ for all $z \in X, u \in U$. Then

1. **Value Iteration:** $V_N = T^N(J_f)$
2. **Bellman Equation:** $V^* \in \mathcal{G}_+$ is a fixed point of $T$, i.e. $T[V^*] = V^*$
3. **Minimum Property:** For any $g \in \mathcal{G}_+$, if $g \succeq T[g]$, then $g \succeq V^*$
4. **Monotonicity:** If $J_f \equiv 0$, then $J_f = V_0 \preceq V_1 \preceq \cdots \preceq V^*$
5. **Stationary Policy:** A stationary policy $\pi_\infty = \{\mu, \mu \ldots\}$ is optimal if and only if $T[V^*] = T_\mu[V^*]$
Properties of Value Iteration Operator III

- Some unfortunate facts:
  - \( \lim_{N \to \infty} V_N \) may not exist

- Even when \( \lim_{N \to \infty} V_N \) exists, we may have \( V_\infty \neq V^* \)

- Bellman equation: \( T[g] = g \) may have infinitely many solutions that are not \( V^* \)

- Counterexamples can be found in [Ber12]

- These cases won’t happen if state space is finite, cost per stage \( l(z, u) \) is bounded, and cost function is discounted.

- However, for control problems, we often don’t have the luxury to make these assumptions
Example 1 (Shortest Path Problem).

Given the road network shown in the figure, find the shortest path from $a_1$ to $a_4$

- $X = \{a_1, \ldots, a_8\}; U(x)$: available links to take at $x \in X$
- dynamics: $x(k+1) = f(x(k+1), u(k+1))$
- running cost: $l(x(k), u(k)) = \text{edge length}$
- terminal cost: $J_f(x) = \begin{cases} 0 & \text{if } x = a_4 \\ \infty & \text{ow} \end{cases}$
- Value iteration:
Examples: Shortest Path Problem II
Example 2 (Viterbi & Hidden Markov Model).

Given a Markov chain with state space $S = \{1, 2, \ldots, n\}$ and transition probability $\{p_{ij}\}_{i,j \leq n}$. Each transition between two states $(i, j)$ produces an independent random observation $z \sim f(z|i, j)$, where $f(z|i, j)$ is the corresponding probability density (or mass) function. Find the most likely state transition sequence $\hat{X}_N = \{\hat{x}_0, \ldots, \hat{x}_N\}$ given an observation sequence $Z = \{z_1, \ldots, z_N\}$, namely, maximize the posterior probability $\text{prob}(X_N|Z_N)$. 
Examples: Viterbi & Hidden Markov Model II
Examples: Linear Quadratic Regulator I

- **$N$-horizon LQR**: Find control sequence $u_0, u_1, ..., u_{N-1}$ to minimize $J_N(z, u)$, subject to linear dynamics constraints:

\[
x_{k+1} = Ax_k + Bu_k, \quad x_0 = z
\]

where: $J_N(x_0, u) = x_N^T Q_f x_N + \sum_{k=0}^{N-1} [x_k^T Q x_k + u_k^T R u_k]$

- **Infinite-horizon LQR**: Find control sequence $u_0, u_1, ...$, to minimize $J_\infty(x_0, u)$ subject to linear dynamics constraints: $x_{k+1} = Ax_k + Bu_k$

where $J_\infty(x_0, u) = \sum_{k=0}^{\infty} [x_k^T Q x_k + u_k^T R u_k]$

- $z^T P z$: quadratic cost term, penalizing deviation from 0, e.g.:
  - if $P = I$, then $z^T P z = \| z \|^2$
  - if $P = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$, then $z^T P z = z_1^2 + 2z_2^2$, penalizes $z_2$ more than $z_1$
Examples: Linear Quadratic Regulator II

Solution of LQR using Dynamic Programming (DP)

- $V_0(z) = z^T Q_f z$

- Suppose at $j$-horizon value function is: $V_j(z) = z^T P_j z$. Compute the $(j+1)$-horizon value function using DP

\[
V_{j+1}(z) = \min_{u \in \mathbb{R}^m} \{l(z, u) + V_j(f(z, u))\}
= \min_{u \in \mathbb{R}^m} \{z^T Q z + u^T R u + (Az + Bu)^T P_j (Az + Bu)\}
= \min_{u \in \mathbb{R}^m} \{u^T(R + B^T P_j B)u + 2z^T A^T P_j B u + z^T(Q + A^T P_j A)z\}
= \min_{u \in \mathbb{R}^m} h(u)
\]

- $\frac{\partial h}{\partial u}(u) = 2u^T(R + B^T P_j B) + 2z^T A^T P_j B = 0$
  \[\Rightarrow\] Optimizer: $\mu^*_j(z) = -(R + B^T P_j B)^{-1} B^T P_j A z \overset{\Delta}{=} -K_{j+1} z$
  where $K_{j+1} = (R + B^T P_j B)^{-1} B^T P_j A$
Examples: Linear Quadratic Regulator III

• Derivation (cont.)

\[ V_{j+1}(z) = \min_{u \in \mathbb{R}^m} h(u) = h(u^*) \]
\[ = (-K_j z)^T (R + B^T P_j B)(-K_j z) + 2z^T A^T P_j B(-K_j z) + z^T (Q + A^T P_j A)z \]
\[ = z^T (Q + A^T P_j A - A^T P_j B (R + B^T P_j B)^{-1} B^T P_j A)z \]
\[ \triangleq z^T P_{j+1} z \]

where \( P_{j+1} \triangleq Q + A^T P_j A - A^T P_j B (R + B^T P_j B)^{-1} B^T P_j A \)

• If the state is \( x_k \) at time \( k \), then the optimal control action applied at time \( k \) is

\[ u_k^* = \mu_{N-k}^*(x_k) = -K_{N-k} x_k \]
Examples: Linear Quadratic Regulator IV

Summary of LQR results:

• $j$-horizon value function: $V_j(z) = z^T P_j z$, where $P_j$ is generated by Riccati recursion

\[
\begin{align*}
  P_{j+1} &= \rho(P_j) \\
  \rho(P) &= Q + A^T PA - A^T PB(R + B^T PB)^{-1} B^T PA
\end{align*}
\] (6)

• To compute LQR controller
  1. Start with $P_0 = Q_f$
  2. Riccati recursion: $P_{j+1} = \rho(P_j)$
  3. Compute optimal feedback gain: $K_{j+1} = -(R + B^T P_j B)^{-1} B^T P_j A$

• To use LQR controller:
  1. initial state: $x_0^*$
  2. compute control input: $u^*(t) = K_{N-t} \cdot x^*(t)$
  3. state update: $x^*(t+1) = Ax^*(t) + Bu^*(t)$
Theorem 2 (Infinite-Horizon LQR).

If \((A, B)\) is stabilizable and \((A, C)\) is detectable, where \(Q = C^T C\), then

- as \(j \to \infty\), \(P_j \to P^*\) and \(K_j \to K^*\) with stable cl-system \((A + BK^*)\);
- \(P^*\) satisfies the algebraic Riccati equation: \(P^* = \rho(P^*)\)
- \(K^* = -(R + B^T P^* B)^{-1} B^T P^* A \triangleq K(P^*)\)

- We will show a more general version of this result (but with simplified setup) later in this lecture

- See [CM70; AM90] for technical details
Outline

- Discrete Time Optimal Control Problem
- Short Introduction to Dynamic Programming
- Connection to Stabilization Problems
Discrete-Time Exponential Stabilization

- $\| \cdot \|$: an arbitrary $p$-norm raised to an arbitrary positive power, namely, $\| \cdot \| \triangleq \| \cdot \|_p^r$, for some $p, r \in \mathbb{R}_+$

**Definition 1 (Exponential stabilizability).**

$\exists \pi_\infty$ such that $\| x(t; z, \pi_\infty) \| \leq c \gamma^t \| z \|$, $\forall z \in X$, $\gamma \in (0, 1)$

**Definition 2 (Exponentially Stabilizing Control Lyapunov Function (ECLF)).**

A PD function $g \in G_+$ is called an ECLF if $\exists \mu$ with $\mu(z) \in U(z)$, $\forall z \in X$, such that

$$
\begin{cases}
\kappa_2 \| z \| \leq g(z) \leq \kappa_1 \| z \| \\
g(z) - g(f(z, \mu(z))) \geq \kappa_3 \| z \|
\end{cases}
$$

(7)
Theorem 3.

Suppose system (1) has ECLF $g$ and let $\mu$ be an arbitrary law satisfying (7). CL-system under $\pi_\infty = \{\mu, \mu, \ldots\}$ is exponentially stable with

$$x(t; z, \pi_\infty) \leq \frac{\kappa_2}{\kappa_1} \left( \frac{1}{1 + \kappa_3/\kappa_2} \right)^t \|z\| \tag{8}$$
Connection Between Optimal Control and Stabilization

How does stabilization relate to optimal control? Let’s start with a well known result for LQR.

**Theorem 4 (LQR solves stabilization problems).**

If \((A, B)\) is stabilizable, and \((A, C)\) is detectable, where \(Q = C^T C\), then

1. Convergence of value iteration: \(P_j \to P^*\) where \(P_{j+1} = \rho(P_j)\) with \(P_0 = 0\)
2. \(V^*(z) = z^T P^* z\) is an ECLF
3. \(\mu(z) = K(P^*) z\) is an stabilizing feedback law

- The theorem indicates: *if we choose cost function properly (i.e. to make \((A, C)\) detectable where \(Q = C^T C\)), then a linear system if stabilizable if and only if it can be stabilized by the LQR controller*

- Question: can we generalize such result? to what extent? and how?
Roughly speaking, *if a system can be (exponentially) stabilized, then we can always stabilize it by solving a properly defined optimal control problem.*

**Key Idea:** select running cost $l : X \times U \to \mathbb{R}_+$ such that

\[
\text{exponentially stabilizable} \Rightarrow V^*(z) \leq \beta \|z\| \quad (9)
\]

\[
V^*(z) \leq \beta \|z\| \Rightarrow \text{the optimal trajectory exponentially stable} \quad (10)
\]

This can be achieved by various types of cost functions. To illustrate the key idea, let us choose the simplest one, say $l(z,u) = \|z\|$.
Theorem 5 (Equivalence between Optimal Control and Stabilization (Simple version)).

Assume zero terminal cost ($J_f \equiv 0$) and a simple running cost function $l(z, u) = \|z\|$. 

1. If system (1) is exponentially stabilizable, then
   
   (i) $V_N \to V^*$ exponentially
   
   (ii) $V^*(z)$ is an ECLF
   
   (iii) The optimal control law $\mu^*$ (i.e. $T_{\mu^*}[V^*] = V^*$), whenever exists, is exponentially stabilizing.

2. If $V^*(z) \leq \beta \|z\|$, then system (1) is exponentially stabilizable and $V^*$ is an ECLF.

Proof:
Connection Between Optimal Control and Stabilization IV

Proof continued
Remarks on Theorem 5

- There are many ways to generalize the results, e.g. nonzero $J_f$, more general running cost functions.

- The main ideas are the same with more involved notations and assumptions (see [Zha09])

- For example, extension to the case with $l(z) = z^T Q z$ would be trivial if assuming $Q \succ 0$.

- This can be further extended by replacing $Q \succ 0$ with some detectability condition (similar to the condition imposed in the LQR theorem). See [AM81; CM70] for classical results on this.
Remarks on Theorem 5

- One can certainly incorporate nontrivial control cost. For example, we can let
  \[ l(z, u) = \|z\| + \|u\| \]

  - This will not affect condition (10), but condition (9) requires more discussions.

  - For example, if the stabilizing control inputs \( u \) does not converge, then \( V^*(z) \) cannot be bounded.

  - All that matters is whether stabilizable with finite "control energy"

  - For linear systems (and many other systems): stabilizability often means stabilizability with finite "control energy"
Conclusion

- Introduced optimal control formulation and dynamic programming approach
- Discussed relation between optimal control and stabilization
- Roughly speaking, if system is exponentially stabilizable, then with properly selected running cost function $l(z,u)$,
  - $V^*$ is an ECLS
  - Optimal infinite-horizon policy $\pi^* = \{\mu^*, \mu^*, \ldots\}$ is exponentially stabilizing
- This provides a unified way to construct ECLF and stabilizing controller.
- However, infinite-horizon optimal control is often more challenging to solve than stabilization
- One remedy is to use $V_N$ (and $\mu_N$) to replace $V^*$ (and $\mu^*$)
  $\Rightarrow$ Model Predictive Control (next lecture)
References


