第十五讲：自适应动态规划

最优控制的智能方法之五

张杰

人工智能学院
中国科学院大学

复杂系统管理与控制国家重点实验室
中国科学院自动化研究所

2017 年 11 月 2 日

Table of Contents

Table of Contents

最优控制问题

问题 (最优控制问题)

1. 被控对象的状态方程为

$$\dot{x}(t) = f(x(t), u(t), t), \; x(t_0) = x_0.$$

2. 容许控制, $u \in U$
3. 目标集, $x(t_f) \in S$
4. 最小化性能指标

$$J(u) = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) dt.$$

离散时间最优控制问题

问题 1 (离散时间最优控制问题)

状态变量为 $x(k) : \mathbb{N} \to \mathbb{R}^n$*，控制变量为* $u(k) : \mathbb{N} \to \mathbb{R}^m$

(1) *被控对象的状态方程*

$$x(k+1) = f_D(x(k), u(k), k), \quad k = 0, \dots, N-1. \tag{1}$$

(2) *容许控制:*

$$u(k) \in U, \quad x(k) \in X. \tag{2}$$

(3) *目标集:*

$$x(N) \in \mathcal{S}. \tag{3}$$

(4) *性能指标:*

$$J(u; x(k), k) = h_D(x(N), N) + \sum_{i=k}^{N-1} g_D(x(i), u(i), i). \tag{4}$$

动态规划方法

离散：Bellman 方程，

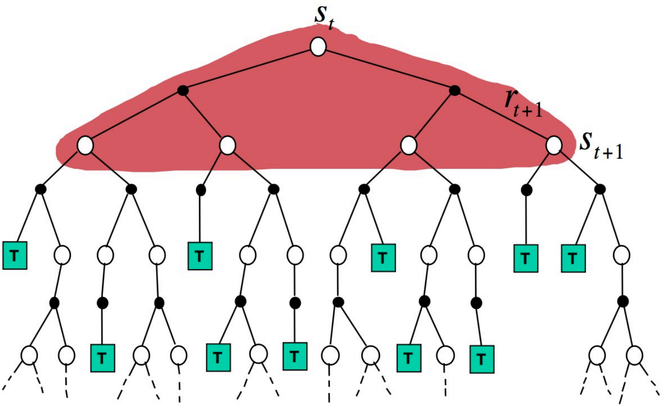
$$V(x(k),k) = \min_{u(k)\in U}\{g_D(x(k),u(k),k) + V(x(k+1),k+1)\}$$
$$k = k_0,\ldots,N-1 \tag{5}$$
$$V(x(N),N) = h_D(x(N),N). \tag{6}$$

连续：HJB 方程，

$$-\frac{\partial V}{\partial t}(x(t),t) = \min_{u(t)\in\mathbb{R}^m} \mathcal{H}(x(t),u(t),\frac{\partial V}{\partial x}(x(t),t),t), \tag{7}$$
$$V(x(t_f),t_f) = h(x(t_f),t_f). \tag{8}$$

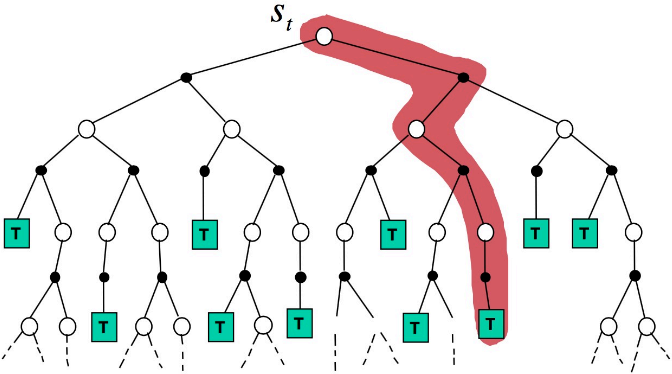
Dynamic Programming

$$V(S_t) \leftarrow E_\pi[R_{t+1} + \gamma V(S_{t+1})]$$



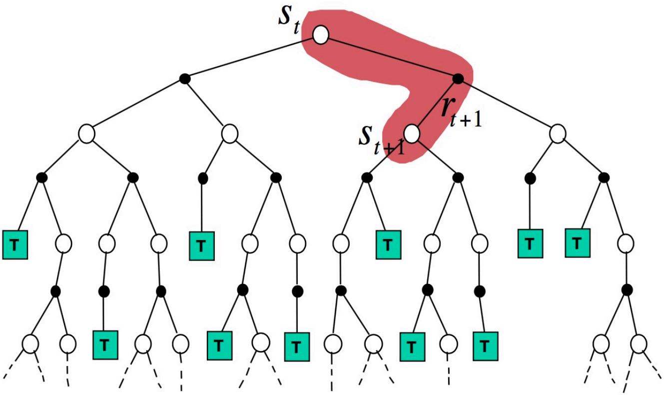
Monte-Carlo

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$$



Temporal-Difference

$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$



课程内容

- 最优控制的数学理论
  - 经典变分法
  - 庞特里亚金极值原理
  - 动态规划方法
  - 微分博弈
- 最优控制的智能方法
  - 模型预测控制
  - 强化学习与自适应动态规划
  - 模糊控制
  - 平行控制与平行学习

参考资料

- Wang, Fei-Yue, Huaguang Zhang, and Derong Liu. "Adaptive dynamic programming: an introduction." Computational Intelligence Magazine, IEEE 4, no. 2 (2009): 39-47.
- Prokhorov, Danil V., and Donald C. Wunsch. "Adaptive critic designs." Neural Networks, IEEE Transactions on 8, no. 5 (1997): 997-1007.
- Lewis, Frank L., and Draguna Vrabie. "Reinforcement learning and adaptive dynamic programming for feedback control." Circuits and Systems Magazine, IEEE 9, no. 3 (2009): 32-50.

Table of Contents

无穷时间最优控制问题

Remark 1

为构造渐进稳定的控制系统，应用中的最优控制终端时间 t_f 或 N 常趋于无穷

- 没有终点 $\Rightarrow$ 无法从终点开始
- 即使找到终点 $\Rightarrow$ 运算无穷次才能到起点 x_0，永远不能完成
- 无法“打靶” $\Rightarrow$ 除可解析求解的问题无法求解 PMP 的 BVP

无穷时间最优控制的最优性原理

Bellman 方程

$$x(k+1) = f_D(x(k), u(k)) \tag{9}$$

$$J(u; x_0) = \sum_{k=0}^{\infty} g_D(x(k), u(k)) \tag{10}$$

$$V(x(k)) = \min_{u(k)}\{g_D(x(k), u(k)) + V(x(k+1))\} \tag{11}$$

HJB 方程

$$\dot{x}(t) = f(x(t), u(t)) \tag{12}$$

$$J(u; x_0) = \int_0^{\infty} g(x(t), u(t))dt \tag{13}$$

$$0 = \min_{u}\{g(x(t), u(t)) + \frac{d}{dx}V(x(t)) \cdot f(x(t), u(t))\} \tag{14}$$

自适应动态规划

自适应动态规划（Adaptive/Approximate Dynamic Programming）由 P. J. Werbos 首次提出

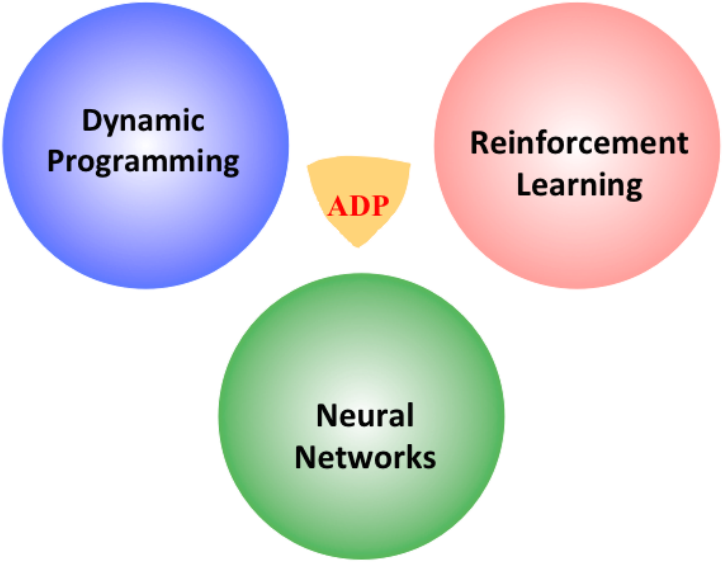
- 利用强化学习的思想
- 结合函数近似结构，逼近性能指标函数和控制策略满足最优性原理
- 时间向前 (Forward-in-time) 获得最优控制

自适应动态规划的名称

- Adaptive Dynamic Programming (ADP)
- Approximate Dynamic Programming (ADP)
- Asymptotic Dynamic Programming (ADP)
- Relaxed Dynamic Programming (RDP)
- Neuro-dynamic Programming (NDP)
- Neural Dynamic Programming (NDP)
- Adaptive Critic Designs (ACDs)

本课中统称 ADP, Adaptive/Approximate Dynamic Programming

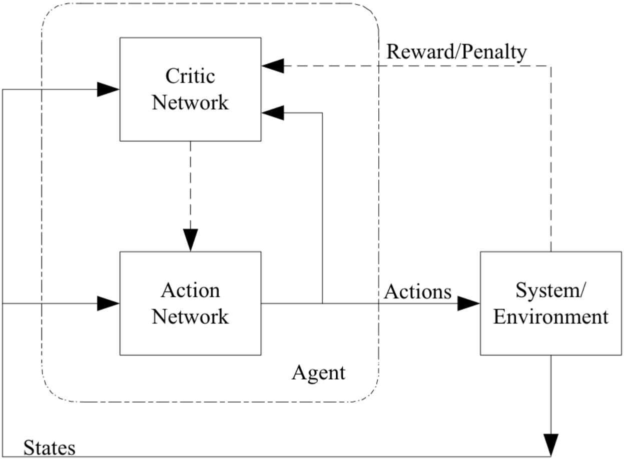
自适应动态规划基本原理



自适应动态规划基本原理

- 整个结构主要由三部分组成:
  - 动态系统（状态方程）
  - 执行函数（控制律）
  - 评判函数（性能指标）
- 每个部分均可由神经网络代替:
  - 动态系统可以通过神经网络进行建模
  - 执行 (Action) 网络用来近似最优控制策略
  - 评判 (Critic) 网络用来近似最优性能指标函数

自适应动态规划基本原理



自适应动态规划基本原理

一般的自适应动态规划方法由三个网络构成, 分别是:

- 模型网络 (Model Network): 输入状态变量和控制变量，输出下时刻状态变量
- 评判网络 (Critic Network): 根据最优性原理，对控制信号进行评价，同时给出评价 (奖励/惩罚) 信号
- 执行网络 (Action Network): 根据 Critic 网络的评价信号更新控制策略，使得 Critic 网络满足最优性原理

自适应动态规划的基本结构

Adaptive Dynamic Programming (ADP)

- HDP: Heuristic dynamic programming
- DHP: Dual heuristic dynamic programming
- GDHP: Globalized DHP

HDP92: Heuristic Dynamic Programming

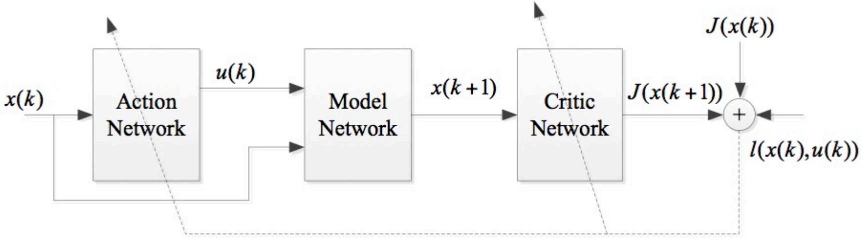


Figure: Werbos, 1992

HDP92: Heuristic Dynamic Programming

目的：满足 Bellman 方程

$$V(x(k)) = \min_{u(k)}\{g_D(x(k), u(k)) + V(x(k+1))\} \tag{15}$$

有

$$V(x(k)) - g_D(x(k), u(k)) - V(x(k+1)) = 0 \tag{16}$$

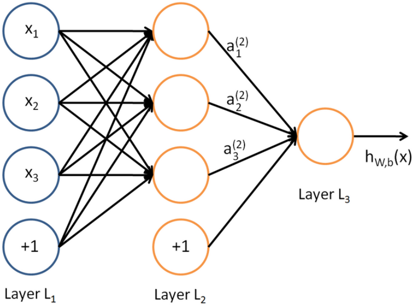
定义目标函数

$$V(x(k+1)) = V(x(k)) - g_D(x(k), u(k)) \tag{17}$$

HDP 输出 $\hat{V}(x(k+1))$，最小化

$$E = \frac{1}{2}[\hat{V}(x(k+1)) - \hat{V}(x(k)) + g_D(x(k), u(k))]^2 \tag{18}$$

多层神经网络



$$
\begin{aligned}
a_1^{(2)} &= f_1(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \\
a_2^{(2)} &= f_1(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \\
a_3^{(2)} &= f_1(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \\
h(x;W,b) = a_1^{(3)} &= f_2(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})
\end{aligned}
$$

HDP 的模型网络 1/3

模型网络输出

$$\hat{x}(k+1) = \omega_m^{(2)} \cdot \tanh(\omega_m^{(1)} \cdot y(k)) \tag{19}$$

$$y(k) := [x^T(k), u^T(k)]^T, \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{20}$$

模型网络应最小化

$$E_m(k) = \frac{1}{2}|\hat{x}(k+1) - x(k+1)|^2 \tag{21}$$

利用误差反向传播算法更新参数。通常充分激励，单独训练

HDP 的 Critic 网络 2/3

Critic 网络输出

$$\hat{V}(x(k)) = \omega_c^{(2)} \cdot \tanh(\omega_c^{(1)} \cdot x(k)) \tag{22}$$

$$V(x(k)) = g_D(x(k), \phi(x(k))) + \hat{V}(x(k+1)) \tag{23}$$

其中 ϕ 为控制策略。Critic 网络最小化

$$E_c(k) = \frac{1}{2}|\hat{V}(x_k) - V(x_k)|^2 \tag{24}$$

利用误差反向传播算法更新参数

HDP 的 Action 网络 3/3

Action 网络输出

$$\phi(x(k)) = \omega_a^{(2)} \cdot \tanh(\omega_a^{(1)} \cdot x(k)) \tag{25}$$

根据最优性原理，控制变量应满足

$$u(k) = \underset{u(k)}{\operatorname{argmin}}\{g_D(x(k), u(k)) + V(x(k+1))\}$$

Action 网络应最小化

$$E_a(k) = \frac{1}{2}|\phi(x(k)) - u(k)|^2 \tag{26}$$

利用误差反向传播算法更新参数

例：HDP 求解无穷时间最优控制问题

例 1

考虑如下非线性离散状态方程

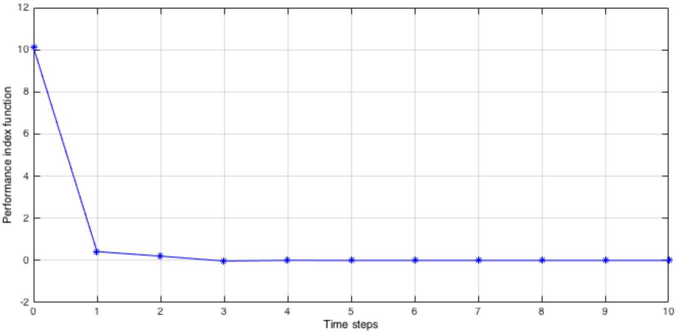
$$x(k+1) = \begin{bmatrix} 0.1x_1^2(k) + 0.05x_2^2(k) \\ 0.2x_2^2(k) - 0.15x_2(k) \end{bmatrix} + \begin{bmatrix} 0.1 + x_1(k) & 0.3 + x_2(k) & 0.5 + x_1(k) \\ 0.3 + x_2^2(k) & 0.1 + x_1^2(k) & 0.3 + x_1(k)x_2(k) \end{bmatrix} u(k) \tag{27}$$

性能指标

$$J = \sum_{k=0}^{\infty} x^T(k)Qx(k) + u^T(k)Ru(k), \tag{28}$$

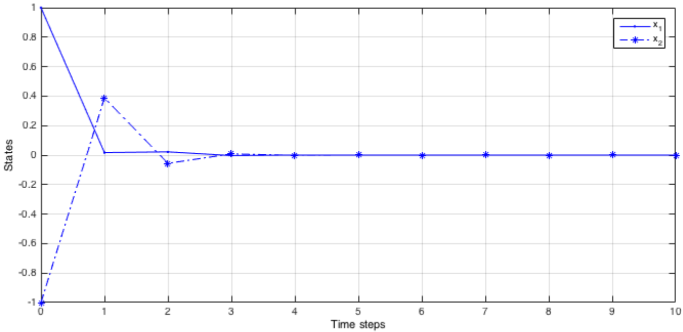
其中 $Q = 0.8I, R = I$

性能指标 1/3



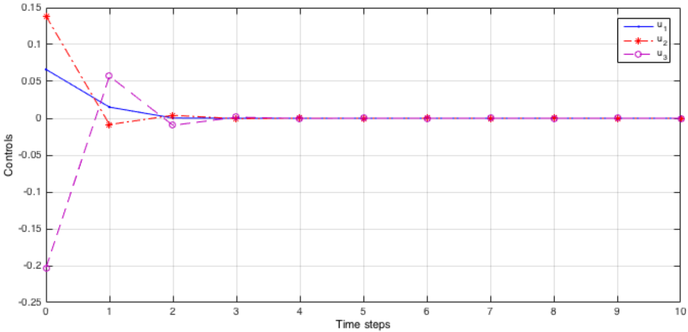
Performance index function
Time steps
12
10
8
6
4
2
0
-2
0
1
2
3
4
5
6
7
8
9
10

状态变量 2/3



States
Time steps
x1
x2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7
8
9
10

控制变量 3/3



0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
Controls
0
1
2
3
4
5
6
7
8
9
10
Time steps
u_1
u_2
u_3

DHP92: Dual Heuristic Dynamic Programming

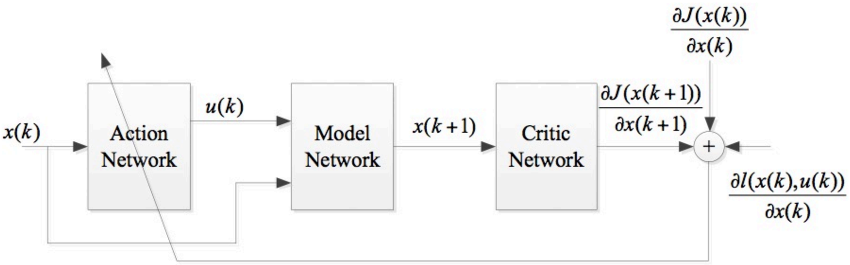


Figure: Werbos, 1992

DHP92: Dual Heuristic Dynamic Programming

目的：满足 Bellman 方程

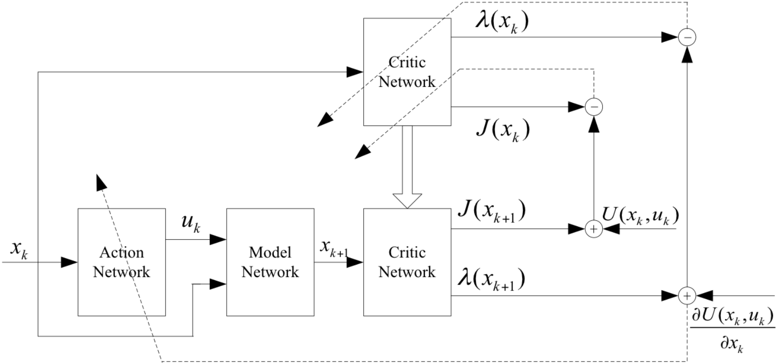
$$V(x(k)) = \min_{u(k)}\{g_D(x(k),u(k)) + V(x(k+1))\} \tag{29}$$

$$\begin{aligned}\frac{\partial V(x(k))}{\partial x(k)} &= \frac{\partial g_D(x(k),u(k))}{\partial x(k)} + \frac{\partial V(x(k+1))}{\partial x(k)} \\ &= \frac{\partial g_D(x(k),u(k))}{\partial x(k)} + \frac{\partial V(x(k+1))}{\partial x(k+1)}\frac{\partial x(k+1)}{\partial x(k)}\end{aligned} \tag{30}$$

DHP 输出 $\partial\hat{V}/\partial x$，最小化

$$E = \frac{1}{2}\left|\frac{\partial \hat{V}(x(k))}{\partial x(k)} - \frac{\partial g_D(x(k),u(k))}{\partial x(k)} - \frac{\partial \hat{V}(x(k+1))}{\partial x(k+1)}\frac{\partial x(k+1)}{\partial x(k)}\right|^2 \tag{31}$$

GDHP: Globalized DHP



GDHP: Globalized DHP

GDHP 的 Critic 网络既逼近性能指标函数又逼近性能指标函数的导数，有两个输出：

$$V(x(k)) \tag{32}$$

$$\frac{\partial V(x(k))}{\partial x(k)} \tag{33}$$

自适应动态规划方法比较

- 结构：简单 $\rightarrow$ 复杂
  HDP $\rightarrow$ DHP $\rightarrow$ GDHP
- 计算精度：高 $\rightarrow$ 低
  GDHP $\rightarrow$ DHP $\rightarrow$ HDP

Table of Contents

迭代自适应动态规划

- 值迭代自适应规划方法
  Value iterative adaptive dynamic programming
- 策略迭代自适应规划方法
  Policy iterative adaptive dynamic programming

值迭代方法

策略迭代需要从一个容许控制策略出发，值迭代方法是一种十分便捷的近似，并不依赖于初始策略，令

$$V_0(x(k)) = 0,\ \forall x(k) \in X.$$

在每次迭代中 $i = 0, 1, \ldots$，首先计算迭代的近似最优控制律，$\forall x(k) \in X$，

$$\phi_i(x(k)) \leftarrow \underset{u(k)\in U}{\operatorname{argmin}} \Big\{ g_D(x(k), u(k)) + V_i(f_D(x(k), u(k))) \Big\}.$$

在此基础上，更新迭代的近似值函数：

$$V_{i+1}(x(k)) \leftarrow g(x(k), \phi_i(x(k))) + V_i(f_D(x(k), \phi_i(x(k)))).$$

$V_0 \leq V_1 \leq \ldots$

Remark 2

2016 年，魏庆来、刘德荣等给出其他初始值函数下的收敛证明

一个最简单的例子

例 2

状态变量 $x(k) : \mathbb{N} \to \mathbb{R}$，控制变量 $u(k) : \mathbb{N} \to \mathbb{R}$。满足离散时间状态方程，

$$x(k+1) = x(k) + u(k).$$

要将状态控制在原点附近并保持稳定。设计二次型性能指标：

$$J(u) = \sum_{k=0}^{\infty} [x^2(k) + u^2(k)]. \tag{34}$$

值迭代方法

在值迭代方法中，对任意的 $x \in \mathbb{R}$，令 $V_0(x) = 0$.
对任意 $x(k) \in \mathbb{R}$,

$$\phi_0(x(k)) \leftarrow \underset{u(k)\in\mathbb{R}}{\operatorname{argmin}} \left\{ x^2(k) + u^2(k) + 0 \right\} = 0.$$

$$V_1(x(k)) \leftarrow \left\{ x^2(k) + \phi_0^2(x(k)) + 0 \right\} = x^2(k).$$

$$\phi_1(x(k)) \leftarrow \underset{u(k)\in\mathbb{R}}{\operatorname{argmin}} \left\{ x^2(k) + u^2(k) + [x(k) + u(k)]^2 \right\} = -0.5x(k).$$

$$V_2(x(k)) \leftarrow \left\{ x^2(k) + \phi_1^2(x(k)) + [x(k) + \phi_1^2(x(k))] \right\} = 1.5x^2(k).$$

不同迭代次数

$i=0$ 时，控制策略非容许

$$u(k)=0.$$

对 $i=1,2,5$，分别有

$$\begin{aligned} u(k) &= -0.5x(k) \\ u(k) &= -0.6x(k) \\ u(k) &= -0.618x(k). \end{aligned}$$

i 再增加，解得的控制策略收敛于 $-0.618x(k)$.

例：值迭代自适应动态规划仿真

例 3 (扭摆系统)

$$\frac{d\theta}{dt} = \omega, \tag{35}$$

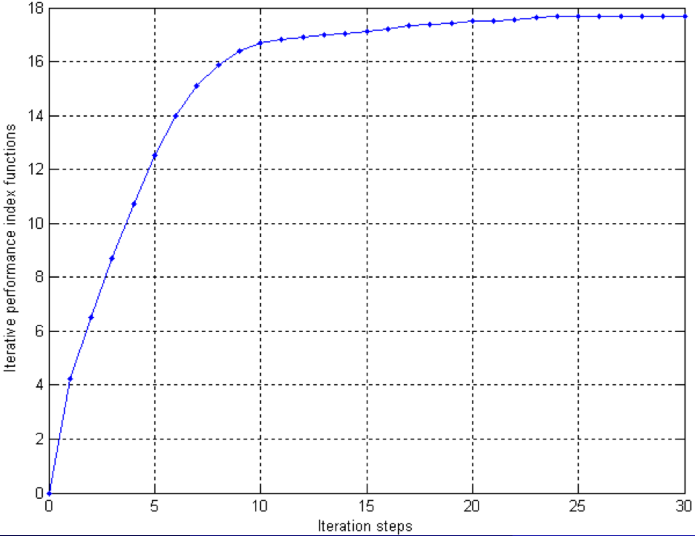
$$J\frac{d\omega}{dt} = u - Mgl\sin\theta - f_d\frac{d\theta}{dt}. \tag{36}$$

θ: 速度; ω: 角速度; $J = 4/3$: 转动惯量; $g = 9.8$; $M = 1/3$: 质量; $l = 2/3$: 摆半径; $f_d = 0.2$: 摩擦系数.

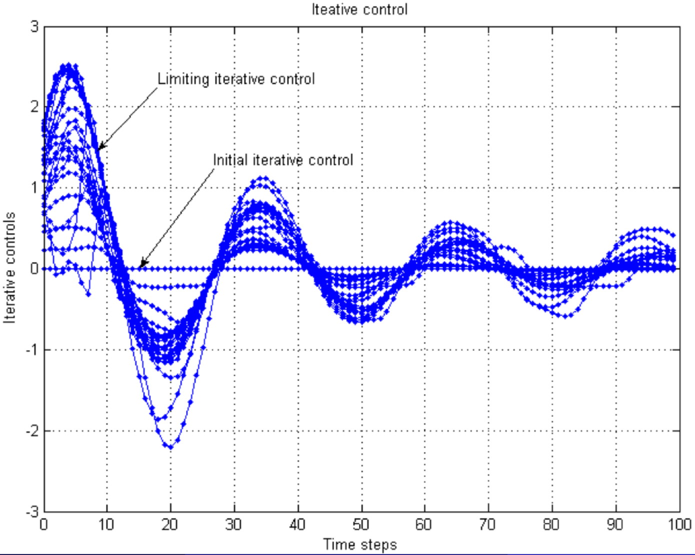
$$J(x_0; u) = \sum_{k=0}^{\infty}(x_k^T Q x_k + u_k^T R u_k), \tag{37}$$

$Q = 0.2I_1, R = 0.2I_2$

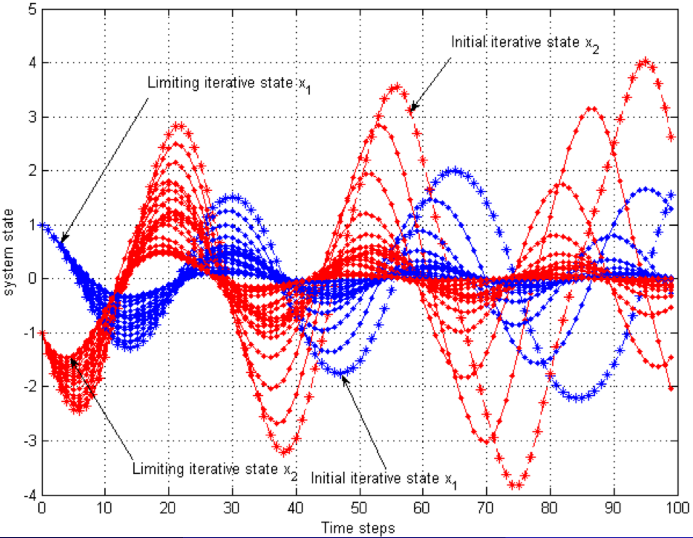
传统值迭代仿真 1/5


Iterative performance index functions
Iteration steps
0
2
4
6
8
10
12
14
16
18
0
5
10
15
20
25
30

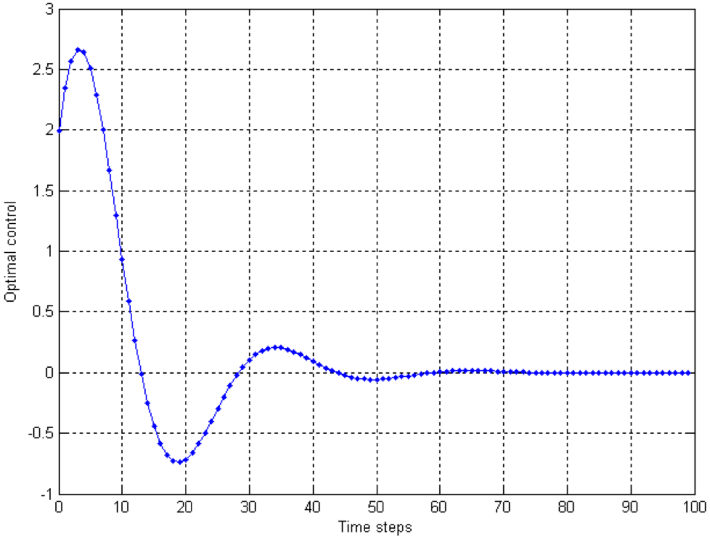
传统值迭代仿真 2/5


Iteative control
Limiting iterative control
Initial iterative control
Iterative controls
Time steps
3
2
1
0
-1
-2
-3
0
10
20
30
40
50
60
70
80
90
100

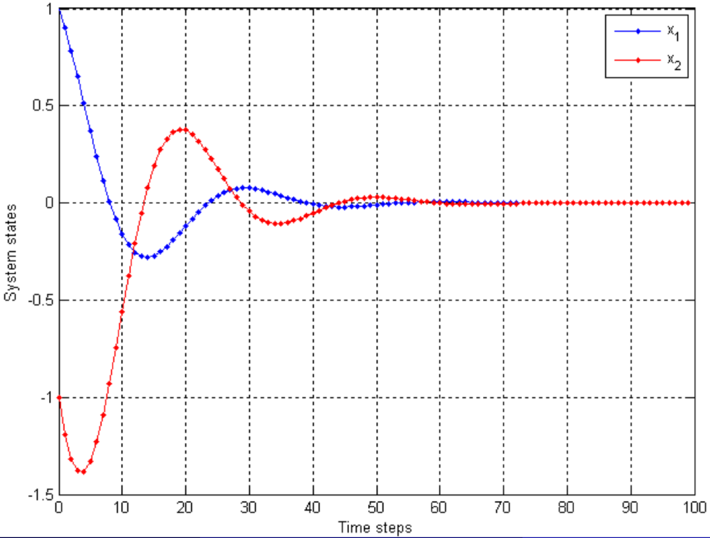
传统值迭代仿真 3/5


Initial iterative state x_2
Limiting iterative state x_1
Limiting iterative state x_2
Initial iterative state x_1
system state
Time steps
5
4
3
2
1
0
-1
-2
-3
-4
0
10
20
30
40
50
60
70
80
90
100

传统值迭代仿真 4/5


Optimal control
Time steps
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
10
20
30
40
50
60
70
80
90
100

传统值迭代仿真 5/5


1
0.5
0
-0.5
-1
-1.5
0
10
20
30
40
50
60
70
80
90
100
System states
Time steps
x_1
x_2

预测模型与滚动优化

$N=2$，在任意时刻 $k=0,1,2,\ldots,$ 分别求解下列最优控制问题：

$$
\begin{aligned}
J(u) = &\sum_{i=k}^{k+N-1} [x^2(i;k)+u^2(i;k)].\\
&x(k;k)=x(k),\\
&x(i+1;k)=x(i;k)+u(i;k), \quad i=k,k+1,\ldots,k+N-1,
\end{aligned}
$$

在时刻 k，首先从环境或仿真中获取状态 $x(k)$。以 $x(i;k)$ 为状态变量，以 $u(i;k)$ 为控制变量，当 $N=2$ 时上述问题可以化简为：

$$
\begin{aligned}
&\min\{x^2(k;k)+u^2(k;k)+x^2(k+1;k)+u^2(k+1;k)\}\\
=&\min\{x^2(k)+u^2(k;k)+[x(k)+u(k;k)]^2+u^2(k+1;k)\}
\end{aligned}
$$

最优解为 $u(k;k)=-x(k)/2$, $u(k+1;k)=0$.

反馈矫正

虽然根据预测模型求得的最优控制为

$$u(k;k) = -x(k)/2, \quad u(k+1;k) = 0.$$

仅实施第一个时段的控制变量，即 $u(k) = -x(k)/2$。
随后反馈矫正，构造预测模型，在本例状态方程不变，初值 $x(k+1;k+1) = x(k+1)$ 可由环境或仿真中观测。滚动优化可得

$$u(k+1) = -x(k+1)/2.$$

$N=2$ 情况下，本例的模型预测控制课解得闭环形式控制策略，

$$\boxed{u(k) = -x(k)/2}$$

不同预测时段的模型预测控制

$N=1$ 时，控制策略非容许

$$u(k)=0.$$

对 $N=2,3,5$，分别得到：

$$\begin{aligned}u(k)=&-0.5x(k)\\u(k)=&-0.6x(k)\\u(k)=&-0.618x(k).\end{aligned}$$

N 再增加，解得的控制策略收敛于 $-0.618x(k)$.

最优性原理与策略迭代

根据最优性原理，有

$$V(x(k)) = \min_{u(k)}\{g_D(x(k), u(k)) + V(x(k+1))\}$$

1960 年，Ronald Howard 提出离散时间系统的策略迭代（policy iteration）：若已知一个容许控制策略 $u(k) = \phi_0(x(k))$，则对 $i = 0, 1, 2, \ldots$，先求解关于 V_i 的广义 *Bellman* 方程：

$$V_i(x(k)) = g_D(x(k), u(k)) + V_i(x(k+1)), \quad \forall x(k) \in X.$$

再解

$$\phi_{i+1}(x(k)) \leftarrow \underset{u(k)}{\operatorname{argmin}}\left\{g_D(x(k), u(k)) + V_i(x(k+1))\right\}, \quad \forall x(k) \in X.$$

可利用神经网络!

离散策略迭代仿真

例 4

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.1x_2(k) + x_1(k) \\ -0.49\sin(x_1(k)) - 0.1f_d \times x_2(k) + x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(k) \tag{38}$$

初始状态为

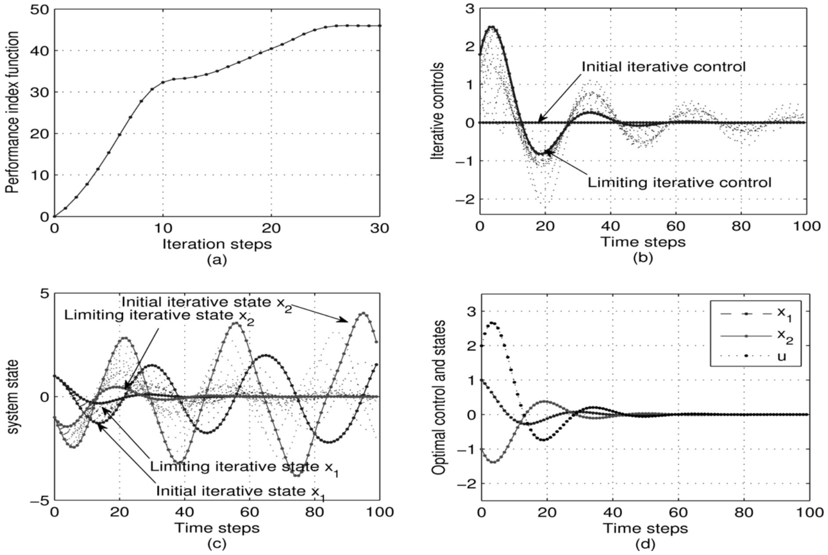
$$x_0 = [1, -1]^T \tag{39}$$

最小化性能指标函数

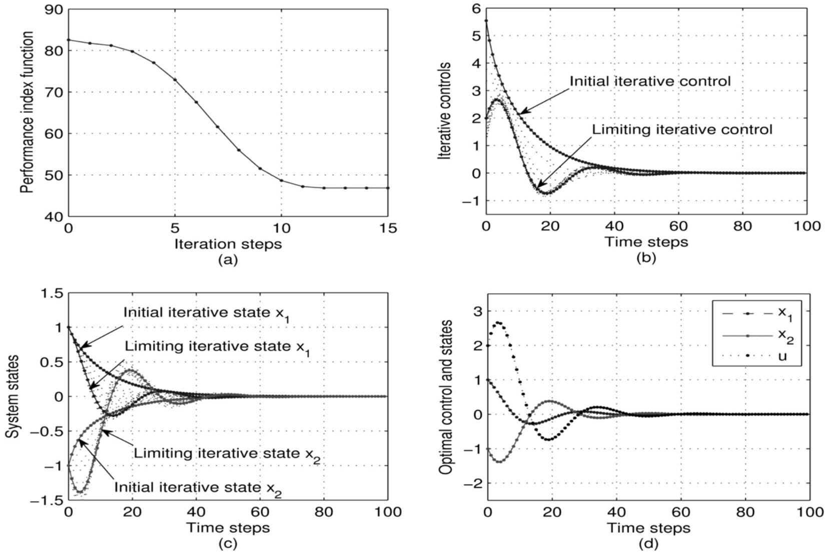
$$J(x_0) = \sum_{k=0}^{\infty} (x^T(k)Qx(k) + u^T(k)Ru(k)) \tag{40}$$

其中 $Q = R = I$.

值迭代仿真 1/2


50
40
30
20
10
0
Performance index function
0
10
20
30
Iteration steps
(a)
3
2
1
0
−1
−2
Iterative controls
Initial iterative control
Limiting iterative control
0
20
40
60
80
100
Time steps
(b)
5
0
−5
system state
Initial iterative state x2
Limiting iterative state x2
Limiting iterative state x1
Initial iterative state x1
0
20
40
60
80
100
Time steps
(c)
3
2
1
0
−1
−2
Optimal control and states
x1
x2
u
0
20
40
60
80
100
Time steps
(d)

策略迭代仿真 2/2


90
80
70
60
50
40
0
5
10
15
Performance index function
Iteration steps
(a)
6
5
4
3
2
1
0
−1
0
20
40
60
80
100
Iterative controls
Initial iterative control
Limiting iterative control
Time steps
(b)
1.5
1
0.5
0
−0.5
−1
−1.5
System states
Initial iterative state x1
Limiting iterative state x1
Limiting iterative state x2
Initial iterative state x2
Time steps
(c)
3
2
1
0
−1
−2
Optimal control and states
x1
x2
u
Time steps
(d)

连续时间自适应动态规划

1979 年，George N. Saridis 提出利用逐次近似近似求解 HJB 方程，是 Howard 离散时间策略迭代的推广。对

$$J(u; x_0, t_0) = h(x(t_f)) + \int_{t_0}^{t_f} \left[L(x(t)) + \|u(t)\|^2\right] \mathrm{d}t, \quad x(t_0) = x_0.$$

$$\dot{x}(t) = A(x(t), t) + B(x(t), t)u(t), \quad t \in [t_0, t_f].$$

已知容许控制 $u(t) = \phi_0(x(t, t))$。$i = 0, 1, \ldots,$ 先解广义 *HJB 方程*:

$$\frac{\partial V_i}{\partial t} + \mathcal{H}(x, u, \frac{\partial V_i}{\partial x}, t) = 0, \quad t \in [t_0, t_f].$$

再迭代控制律:

$$\phi_{i+1}(x, t) \leftarrow -\frac{1}{2} B^{\mathrm{T}}(x, t) \frac{\partial V_i}{\partial x}(x, t), \quad i = 1, 2, \ldots$$

有 $V_0(x, t) \geq V_1(x, t) \geq \ldots.$

连续时间策略迭代自适应动态规划

考虑连续最优控制。状态方程

$$\dot{x} = A(x) + B(x)u \tag{41}$$

$$J(x) = \int_0^\infty g(x,u)dt = \int_0^\infty (x^T Qx + u^T Ru)dt \tag{42}$$

$$V(x) = \min J(u;x) \tag{43}$$

HJB 方程为

$$0 = \min_u \{g(x,u) + \frac{dV(x)}{dx} \cdot [A(x) + B(x)u]\} \tag{44}$$