Advances in Applied Mathematics 应用数学进展, 2022, 11(12), 8836-8845
Published Online December 2022 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2022.1112931

Hans汉斯

带熵的随机线性二次最优控制问题

舒　心

上海理工大学，理学院，上海

收稿日期：2022年11月21日；录用日期：2022年12月15日；发布日期：2022年12月23日

摘　要

本文研究了随机线性二次最优控制问题，对带有随机参数的无限时间内的离散时间线性二次最优控制问题，我们不考虑控制过程本身的最优解而是求解控制过程的概率分布，并用熵来度量这个随机概率分布的探索水平。经计算得到控制过程的最优概率分布服从高斯分布，再利用概率分布可求得线性二次型最优控制问题值函数的各项系数矩阵的迭代式。在值迭代的基础上使用Q-learning算法求解各项系数值的平稳解。最后选择两个数值算例证明了Q-learning算法的有效性，并比较了加熵和不加熵时的算法效果，结果表明熵的运用可以使算法收敛更快更稳定。

关键词

随机线性二次最优控制，熵，概率分布

Linear Quadratic Optimal Control Problem with Entropy

Xin Shu

College of Science, University of Shanghai for Science and Technology, Shanghai

Received: Nov. 21st, 2022; accepted: Dec. 15th, 2022; published: Dec. 23rd, 2022

Abstract

This paper studies the stochastic linear quadratic optimal control problem. For the discrete time linear quadratic optimal control problem with random parameters in infinite time, we do not consider the optimal solution of the control process itself, but solve the probability distribution of the control process, and use entropy to measure the exploration level of this stochastic probability distribution. The calculation results show that the optimal probability distribution of the control

文章引用: 舒心. 带熵的随机线性二次最优控制问题[J]. 应用数学进展, 2022, 11(12): 8836-8845.
DOI: 10.12677/aam.2022.1112931

process obeys the Gaussian distribution. By using the probability distribution, the iterative formulas of the coefficients of the linear quadratic optimal control problem value function can be obtained. According to the value iteration, Q-learning algorithm is used to solve the stationary solution of each coefficient value. Finally, two numerical examples are selected to illustrate the effectiveness of Q-learning algorithm, and the effect of the algorithm with and without entropy is compared. The results show that the application of entropy can make the algorithm convergence faster and more stable.

Keywords

Stochastic Linear Quadratic Optimal Control, Entropy, Probability Distribution

1. 引言

线性二次最优控制问题具有非常广泛的应用，在金融、工程等领域有很大的实用价值。Kamlman 首先建立了确定性线性最优二次控制理论[1]，Wonham 将此类问题扩展到具有可观察性的确定性参数问题[2] [3]，Bismut 又将问题扩展到参数随机的随机线性二次控制最优问题[4]。自此，确定性和随机性线性二次最优控制问题被广泛研究。

本文研究的就是随机线性二次最优控制问题。随机线性二次控制优化问题自提出以来，在理论和应用上都得到了快速的发展。Prozaton 等人考虑一类参数未知的离散线性系统的二次最优控制问题，利用信息矩阵预测参数的后验不确定性，即后验分布的协方差矩阵，设计了主动自适应控制策略[5]。在确定性线性二次问题中，成本函数中对应的控制权重矩阵是正定的时候可以通过黎卡提方程求解，很多研究者在研究随机线性最优控制问题时都会假设控制权重矩阵非负定。Chen 等人发现对于一些随机线性最优控制问题，即使控制权重项是负的也可以是 well-posed，并提出了倒向随机微分方程式的非线性随机黎卡提方程，解决了当权重矩阵是非负定以及不确定情况下的问题[6] [7]。Rami 等人解决了当成本函数中的状态和控制权重矩阵不确定(indefinite)时的有限时间内随机线性二次最优控制问题，通过对原有黎卡提方程放松约束条件和增加新的约束条件引入广义黎卡提方程，并构建出最优控制和最优成本值[8]。Wang 等人基于值迭代使用自适应动态规划算法解决无限时间内的系统动态未知的随机线性二次最优控制问题，并用神经网络识别未知的系统动态以及近似值函数[9]。Du 等人研究了无限时间内随机参数独立同分布的离散线性二次最优控制问题，提出了一种基于 Q-learning 的迭代算法[10]。

概率论中熵是不确定性的度量，不确定性越大熵越大。目前在很多方法中都运用了熵。比如强化学习问题中一些方法在目标函数中加上熵。加熵后的强化学习改变了目标函数，但是在探索性和鲁棒性方面提供了实质性改进。比如 Ziebart 等人所讨论的最大熵策略在面对模型和估计误差时是鲁棒的[11]。Boularias 等人在最大熵框架的基础上介绍了一种新的无模型逆强化学习算法，通过梯度下降方法最小化相对熵来设计算法，并证明了这种方法和已有方法相比有所改进[12]。Haarnoja 等人在最大熵框架下通过获得不同的行为来改进探索[13] [14]。Zhao 等人首先提出了一种基于加权熵的新型多目标强化学习，并开发了一个基于最大熵的优先排序框架来优化所提出的目标。实验显示这种方法在性能和样本效率方面

有很好的改进[15]。Wang 等人提出并发展了一种广义熵正则化、松弛的随机控制公式，称之为探索性公式，以明确捕捉强化学习中探索(exploration)和开发(exploitation)之间的权衡。在他的随机控制公式里智能体随机化它的控制来探索和学习环境，经典的控制被控制的分布取代[16]。同样地，Wang 等人在强化学习框架下解决有限时间内连续时间均值方差投资组合问题的时候引入熵来增加探索率，通过探索权重来反应探索和开发之间的权衡，最终实现最佳交易问题[17]。

在已有的随机线性二次最优控制问题方法的基础上，本文引入熵。对于随机线性二次最优控制问题，目前大部分方法都是根据黎卡提方程或者其他方法直接求最优控制，而本文考虑控制过程的概率分布，即通过求解控制过程的最优概率分布来确定策略，将最优概率分布带入目标函数即可求得随机线性二次型问题各项系数迭代式。数值分析证明了熵的加入使算法收敛更快更稳定。

2. 方法

对于离散时间随机线性二次控制问题，给定初始状态 $x_0 = x \in \mathbb{R}^n$，系统为

$$x_{t+1} = A_{t+1}x_t + B_{t+1}u_t = \Lambda_{t+1}\begin{bmatrix} x_t \\ u_t \end{bmatrix}, t = 0,1,2,\cdots, \tag{2.1}$$

其中 $x_t \in \mathbb{R}^n$ 代表 t 时刻的状态，$u_t \in \mathbb{R}^m$ 代表 t 时刻的控制。现在考虑折算(discount)问题，其中折现因子 $\gamma < 1$，对应成本函数为

$$J(x,u_{tr}) = \sum_{t=0}^{\infty}\gamma^t\left[x_t^{\mathrm{T}}, u_t^{\mathrm{T}}\right]N_{t+1}\begin{bmatrix} x_t \\ u_t \end{bmatrix} \tag{2.2}$$

其中 u_{tr} 代表控制 u 的整个轨迹，即 $\{u_t\}_{t=0}^{\infty}$。对应的价值函数定义为

$$V(x) := \min_{u_{tr}} E\left[J(x,u_{tr})\right] \tag{2.3}$$

其中 Λ_{t+1} 和 N_{t+1} 是随机项，E 表示对随机项求期望。

控制过程 u_t 是随机的，可表示为探索和学习，是一个测量值或者分布控制过程，它的密度函数表示为 $\pi_t(u)$。根据贝尔曼最优原则，值函数可以表示为以下形式

$$V^{\pi}(x) = \min_{\pi\in\mathcal{A}(x)}\int\pi_t(u)E\left[r_t(x,u) + \gamma V^{\pi}(x_{t+1}) \mid x_t = x\right]\mathrm{d}u \tag{2.4}$$

其中 $r(x,u) = \left[x^{\mathrm{T}}, u^{\mathrm{T}}\right]N_{t+1}\begin{bmatrix} x \\ u \end{bmatrix}$ 是在 t 时刻控制概率为 $\pi_t(u)$ 下的即时奖励，$\mathcal{A}(x)$ 是 $x_t = x$ 下的可行性控制分布集(admissible distributional controls)。随机控制过程的概率分布分布可以用 $\pi_t(u)$ 来衡量探索率，可以通过熵来计算它的水平

$$\mathcal{H}(\pi) = -\int\pi_t(u)\ln\pi_t(u)\mathrm{d}u \tag{2.5}$$

考虑熵的问题，探索权重用 $\lambda > 0$ 表示可以权衡探索和利用(exploitation and exploration)。此时值函数变为

$$V^{\pi}(x) = \min_{\pi\in\mathcal{A}(x)}\int\pi_t(u)E\left[r_t(x,u) + \lambda\ln\pi_t(u) + \gamma V^{\pi}(x_{t+1}) \mid x_t = x\right]\mathrm{d}u \tag{2.6}$$

如果值函数是有界的，值函数应该是一个二次型形式，假设值函数 $V^{\pi}(x) = x^{\mathrm{T}}K_2x + K_1x + K_0$，$K_2$ 是半正定矩阵，根据贝尔曼最优原理

$$
\begin{aligned}
&x^{\mathrm{T}}K_2x+K_1x+K_0\\
&=\min_{\pi\in\mathcal{A}(x)}\int\pi_t(u)E\left\{\begin{bmatrix}x^{\mathrm{T}},u^{\mathrm{T}}\end{bmatrix}N_{t+1}\begin{bmatrix}x\\u\end{bmatrix}+\lambda\ln\pi_t(u)+\gamma x_{t+1}^{\mathrm{T}}K_2x_{t+1}+\gamma K_1x_{t+1}+\gamma K_0\mid x_t=x\right\}\mathrm{d}u\\
&=\min_{\pi\in\mathcal{A}(x)}\int\pi_t(u)E\left\{\begin{bmatrix}x^{\mathrm{T}},u^{\mathrm{T}}\end{bmatrix}\left(N_{t+1}+\Lambda_{t+1}^{\mathrm{T}}\gamma K_2\Lambda_{t+1}\right)\begin{bmatrix}x\\u\end{bmatrix}+\lambda\ln\pi_t(u)+\gamma K_1\Lambda_{t+1}\begin{bmatrix}x\\u\end{bmatrix}+\gamma K_0\mid x_t=x\right\}\mathrm{d}u\\
&=\min_{\pi\in\mathcal{A}(x)}\int\pi_t(u)\left(\begin{bmatrix}x^{\mathrm{T}},u^{\mathrm{T}}\end{bmatrix}E\left(N_{t+1}+\gamma\Lambda_{t+1}^{\mathrm{T}}K_2\Lambda_{t+1}\right)\begin{bmatrix}x\\u\end{bmatrix}+E\left(\gamma K_1\Lambda_{t+1}\right)\begin{bmatrix}x\\u\end{bmatrix}+\gamma K_0+\lambda\ln\pi(u)\right)\mathrm{d}u
\end{aligned}
\tag{2.7}
$$

若 $d=n+m$，对于维数为 $d\times d$ 的半正定矩阵 P 和维数为 $1\times d$ 的矩阵 Q，根据下面的划分来引用某些子矩阵

$$
P=\begin{bmatrix}P_{xx} & P_{xu}\\ P_{ux} & P_{uu}\end{bmatrix}\ \text{with}\ P_{xx}\in\mathbb{R}^{n\times n}
\tag{2.8}
$$

$$
Q=\begin{bmatrix}Q_{xx} & Q_{uu}\end{bmatrix}\ \text{with}\ Q\in\mathbb{R}^{n\times n}
\tag{2.9}
$$

并定义映射：

$$
\Pi(P):=P_{xx}-P_{xu}P_{uu}^{+}P_{ux}
\tag{2.10}
$$

$$
\Gamma1(P,Q):=Q_{xx}-Q_{uu}P_{uu}^{+}P_{ux}
\tag{2.11}
$$

$$
\Gamma0(P,Q):=-\frac{1}{4}Q_{uu}P_{uu}^{+}Q_{uu}^{\mathrm{T}}++\frac{\lambda^{m+1}}{2^{m+1}}\ln\left((2\pi e)^m\det\left(P_{uu}^{+}\right)\right)
\tag{2.12}
$$

其中 P_{uu}^{+} 表示 P_{uu} 的 Moore-Penrose 伪逆。令 $M(K_2)=E\left(N_{t+1}+\gamma\Lambda_{t+1}^{\mathrm{T}}K_2\Lambda_{t+1}\right)$，$M(K_2)$ 表示矩阵 K_2 的方程。令 $V(K_1)=E\left(\gamma K_1\Lambda_{t+1}\right)$，$V(K_1)$ 表示矩阵 K_1 的方程。使用上面的符号，并考虑到 $\pi_t(u)\in\mathcal{P}(\mathbb{R})$，即 $\int\pi_t(u)\mathrm{d}u=1$，使用 Gateaux 导数，可以计算出 t 时刻控制过程 u_t 的最优概率 $\pi_t(u)$，最终算得最优概率分布为

$$
\begin{aligned}
\pi^*(u)&=\frac{\exp\left\{-\frac{1}{\lambda}\left(u^{\mathrm{T}}M_{uu}u+u^{\mathrm{T}}M_{ux}x+x^{\mathrm{T}}M_{xu}u+x^{\mathrm{T}}M_{xx}x+V_{uu}u+V_{xx}x+K_0\right)\right\}}{\int\exp\left\{-\frac{1}{\lambda}\left(u^{\mathrm{T}}M_{uu}u+u^{\mathrm{T}}M_{ux}x+x^{T}M_{xu}u+x^{\mathrm{T}}M_{xx}x+V_{uu}u+V_{xx}x+K_0\right)\right\}\mathrm{d}u}\\
&=\mathcal{N}\left(u\mid-\frac{M_{uu}^{+}}{2}\left(2M_{ux}x+V_{uu}^{\mathrm{T}}\right),\frac{\lambda}{2}M_{uu}^{+}\right)
\end{aligned}
\tag{2.13}
$$

把 $\pi^*(u)$ 带回方程(2.7)，可以得到

$$
K_2=\Pi\left(E\left(N_{t+1}+\gamma\Lambda_{t+1}^{\mathrm{T}}K_2\Lambda_{t+1}\right)\right)
\tag{2.14}
$$

$$
K_1=\Gamma1\left(E\left(N_{t+1}+\gamma\Lambda_{t+1}^{\mathrm{T}}K_2\Lambda_{t+1}\right),E\left(\gamma K_1\Lambda_{t+1}\right)\right)
\tag{2.15}
$$

$$
K_0=\frac{1}{1-\gamma}\Gamma0\left(E\left(N_{t+1}+\gamma\Lambda_{t+1}^{\mathrm{T}}K_2\Lambda_{t+1}\right),E\left(\gamma K_1\Lambda_{t+1}\right)\right)
\tag{2.16}
$$

由此我们可以求得控制过程的最优概率分布以及各项系数的迭代式。现在的问题是当参数是随机的时候如何求得随机线性二次最优控制问题。对于最优控制问题，Bertsekas 在他的书里提到了很多强化学习方法[18]。考虑值迭代的形式，选择使用 Q-learning 方法。Q-learning 算法由 Watkins 首次提出，是一种基于值的强化学习方法[19]。一般 Q-learning 算法中的 Q 值 $q(x,u)$ 是关于状态和控制的值，且不容易

直接得到，需要用其他方法近似，而本文内我们不直接求状态和控制的 Q 值而是求了对应值函数的矩阵系数并得到了确定的迭代式，考虑使用 Q-learning 算法思想，所以仍用 Q 表示所求值，根据迭代公式定义函数

$$Q_2^* = E\left[N + \gamma\Lambda^{\mathrm{T}} K_2 \Lambda\right] \tag{2.17}$$

$$Q_1^* = E\left[\gamma K_1 \Lambda\right] \tag{2.18}$$

由此可得

$$Q_2^* = E\left[N + \gamma\Lambda^{\mathrm{T}}\Pi\left(Q_2^*\right)\Lambda\right] \tag{2.19}$$

$$Q_1^* = E\left[\gamma\Gamma 1\left(Q_2^*, Q_1^*\right)\Lambda\right] \tag{2.20}$$

所以最终 $K_2 = \Pi\left(Q_2^*\right)$，$K_1 = \Gamma 1\left(Q_2^*, Q_1^*\right)$。$K_2$ 和 K_1 是 Q_2 和 Q_1 矩阵的子矩阵划分，因此可以先求得 Q_2 和 Q_1，由此设置算法 1 (表 1)：

Table 1. Q-learning algorithm for stochastic linear quadratic optimal control
表 1. 随机线性二次最优控制的 Q-learning 算法

算法 1：随机线性二次最优控制的 Q-learning 算法
1) 初始化矩阵使 $Q_2(0) = I_d$，$Q_1(0) = [1,\cdots,1]_{1\times d}$
2) for $t = 0, T$：
3) $Q_2(t+1) = Q_2(t) + \alpha\left[E\left(N + \gamma\Lambda^{\mathrm{T}} Q_2(t)\Lambda\right) - Q(t)\right]$
4) $Q_1(t+1) = Q_1(t) + \alpha\left[E\left(\gamma\Gamma 1\left(Q_2(t), Q_1(t)\right)\Lambda\right) - Q_1(t)\right]$
5) $K_0(t) = \dfrac{1}{1-\gamma}\Gamma 0\left(Q_2(t), Q_1(t)\right)$
6) $t = t+1$
end for

其中学习率满足 $\sum_{t=0}^{\infty}\alpha_t = +\infty$，$\sum_{t=0}^{\infty}\alpha_t^2 < +\infty$。由于参数是随机的，所以对于算法中的期望部分使用蒙特卡洛方法计算，即 $\frac{1}{S}\sum_{k=1}^{s}\left[N_k + \gamma\Lambda_k^{\mathrm{T}} Q_2(t)\Lambda_k\right]$。

3. 数值分析

为了验证算法的可行性和有效性，进行数值分析。目前已知所求值的迭代公式，所以可以根据迭代公式直接求目标值，因此先比较根据迭代公式求解和根据 Q-learning 算法思路求解的不同。两种方法结果如图。

根据图 1 可以看出直接根据迭代公式用迭代法求解的目标值一直有较大的波动，且在迭代次数范围内并不收敛。而本文选择的 Q-learning 算法求得的目标值比迭代法求得的目标值更准确且最终收敛。由此可以说明本文采用的 Q-learning 算法有一定的优势且更有效。

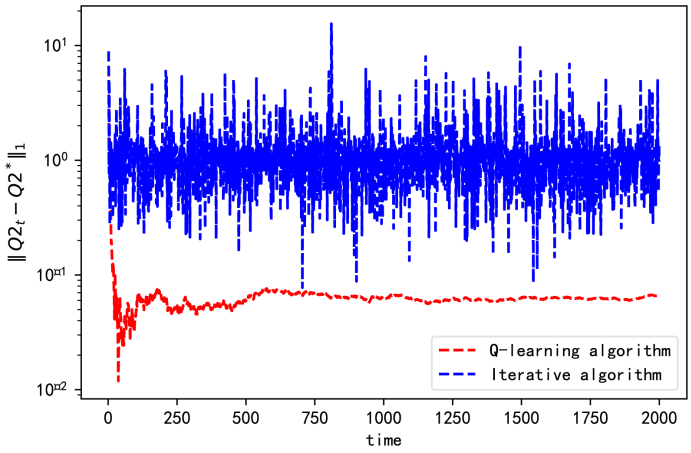

Figure 1. The comparison of iterative algorithm and Q-learning algorithm
图 1. 迭代算法和 Q-learning 算法的比较

本文和其他方法最大的区别在于在目标函数中加入了熵，因此在数值分析过程中主要考虑值函数加熵和不加熵即 $\lambda \neq 0$ 和 $\lambda = 0$ 两种情况，对于 $\lambda \neq 0$ 时需要考虑 λ 取不同值时的情况。比较 $\lambda = 1$、$\lambda = 5$ 和 $\lambda = 10$ 时 Q_2 的收敛情况，

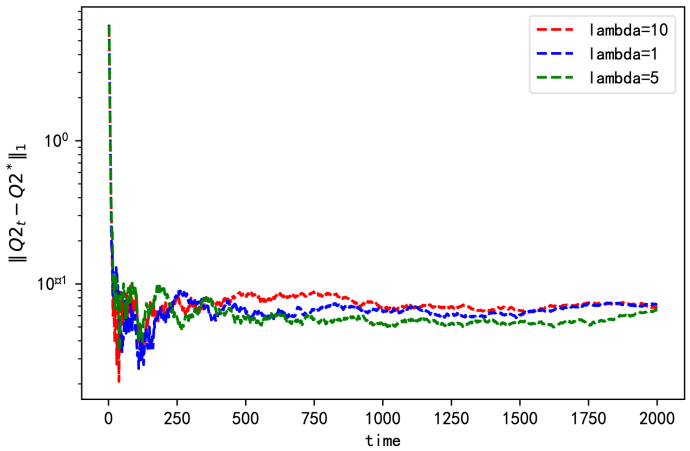

Figure 2. The convergence with different explore weight
图 2. 探索权重不同时的收敛情况

由图 2 可知选取的 3 个不同的 λ 对应的收敛以及稳定情况大致相同，最后选择 $\lambda = 1$。令折扣因子 $\gamma = 0.99$，算法中的学习率 $\alpha = \frac{t}{5+t}$，在蒙特卡洛计算期望值是使 $s = 200$。

首先考虑状态空间 $n = 2$，控制空间 $m = 1$ 时的情况。此时使 $\Lambda_t = \Lambda^{(0)} + \omega_t^{(1)}\Lambda^{(1)} + \omega_t^{(2)}\Lambda^{(2)}$，$N_t = N^{(0)}$，其中 $\omega_t^{(1)}$，$\omega_t^{(2)}$ 独立同分布且服从标准正态分布。

$$\Lambda^{(0)} = \begin{bmatrix} -1 & -0.1 & -0.2 \\ 2.6 & 0.5 & 0.5 \end{bmatrix},\quad \Lambda^{(1)} = \begin{bmatrix} 0.6 & 0.075 & 0.125 \\ -0.8 & 0.1 & -0.375 \end{bmatrix}$$

$$\Lambda^{(2)} = \begin{bmatrix} -0.06 & -0.06 & 0.02 \\ 0.2 & 0.23 & -0.09 \end{bmatrix},\quad N^{(0)} = \begin{bmatrix} 3.11 & 1.5626 & -0.2798 \\ 1.5626 & 1.816175 & -1.021425 \\ -0.2798 & -1.021425 & 0.91585 \end{bmatrix}$$

得到各迭代值收敛情况为(图 3~5)。

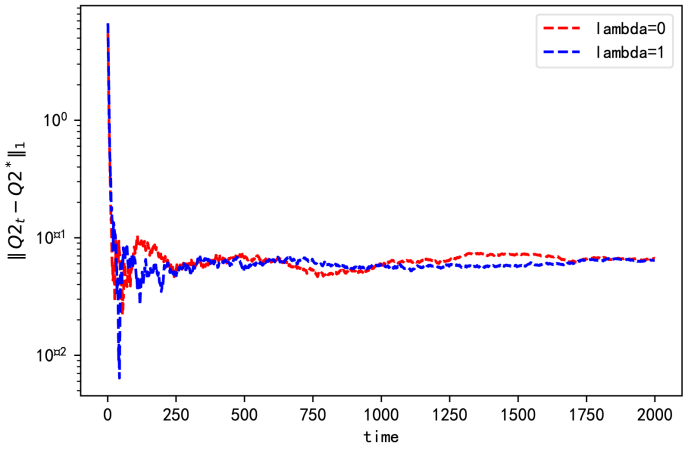

Figure 3. The convergence of Q_2 when $n = 2$
图 3. $n = 2$ 时 Q_2 的收敛情况

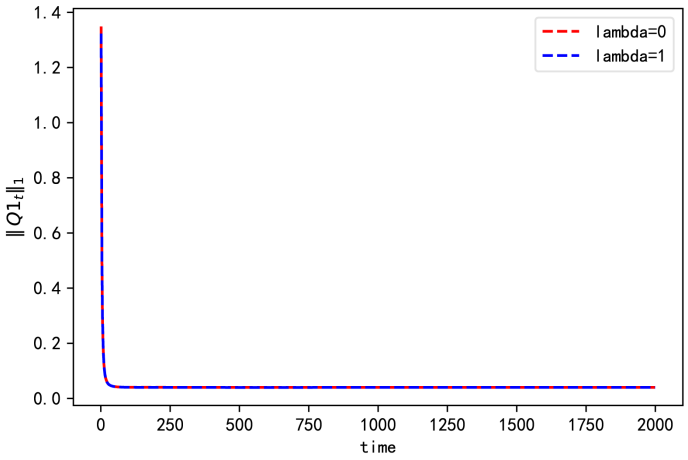

Figure 4. The convergence of Q_1 when $n = 2$
图 4. $n = 2$ 时 Q_1 的收敛情况

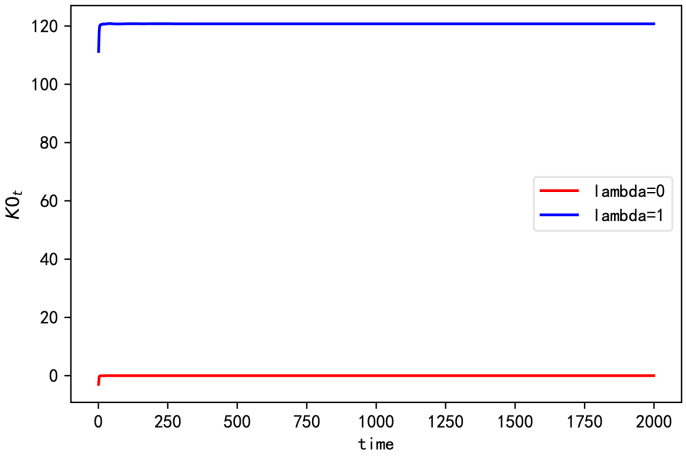

Figure 5. The convergence of K_0 when $n = 2$
图 5. $n = 2$ 时 K_0 的收敛情况

然后考虑状态空间 $n=3$，控制空间 $m=1$ 时的情况，令 $\Lambda_t=\Lambda^{(0)}+\omega_t^{(1)}\Lambda^{(1)}$，其中 $\omega_t^{(1)}$ 标准正态分布。

$$\Lambda^{(0)}=\begin{bmatrix}-0.7718 & 0.3632 & 0.1619 & 0.7298\\ 0.0335 & 0.1955 & -0.0709 & 0.3275\\ -0.0738 & 0.2609 & 0.5275 & -0.5730\end{bmatrix},$$

$$\Lambda^{(1)}=\begin{bmatrix}-0.4505 & 0.0671 & 0.1783 & 0.1651\\ -0.900 & -0.0628 & -0.1045 & -0.4122\\ -0.6539 & -0.4185 & -0.2444 & 0.9814\end{bmatrix},$$

$$N^{(0)}=I_d$$

得到各迭代值收敛情况为(图 6~8)。

由一次项和常数项系数值可以看出，此时在值函数中加熵不会影响他们的收敛情况，但是加熵后存在数值差异，且集中在常数项。由图 3 和图 6 可知，加熵后对应的二次项系数收敛更快更稳定。

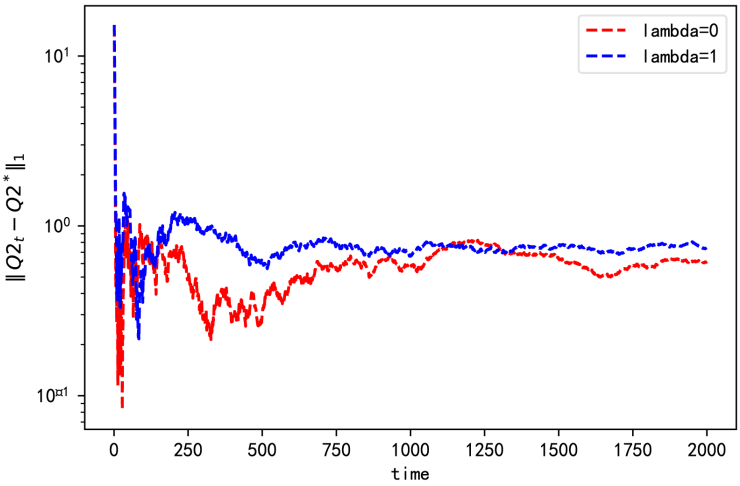

Figure 6. The convergence of Q_2 when $n=3$
图 6. $n=3$ 时 Q_2 的收敛情况

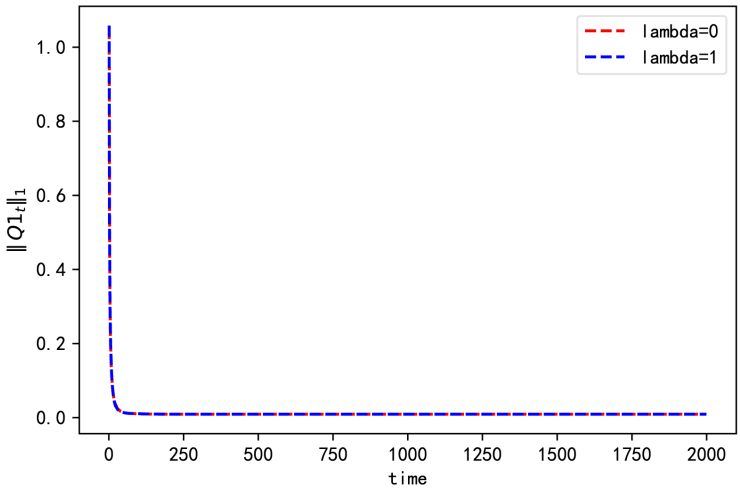

Figure 7. The convergence of Q_1 when $n=3$
图 7. $n=3$ 时 Q_1 的收敛情况

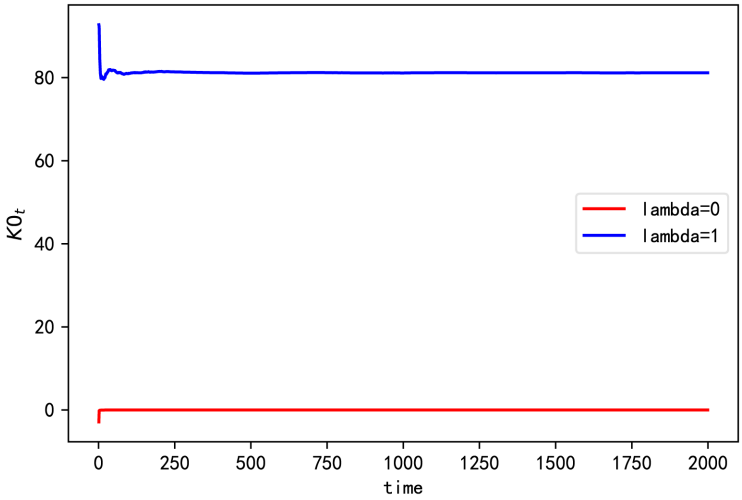

Figure 8. The convergence of K_0 when $n = 3$
图 8. $n = 3$ 时 K_0 的收敛情况

4. 结论

对于确定性和随机性线性二次最优控制问题，目前很多方法都是围绕着黎卡提方程求解最优控制过程，本文不直接考虑控制过程，而是考虑控制过程的概率分布，同时用熵来表示控制过程的探索水平。在原来的目标函数上加熵，对加熵后的目标函数关于概率分布求 Gateaux 导数得到控制过程的最优概率分布，并化简得到控制过程最优概率分布服从高斯分布。将概率分布带回加熵后的目标函数求出线性二次函数的各系数的迭代公式，使用蒙特卡洛算法计算期望值，最后用 Q-learning 算法得到各系数的最终平稳值。比较直接用迭代公式求解和用本文使用的 Q-learning 算法求解的结果，证明了本文算法有一定的优势且更有效，两个数值分析证明了算法的收敛性，同时表明了加熵后改变了目标函数值的大小，但是差值主要集中在常数项，也证明了熵的运用使算法收敛更快更稳定。

致　谢

感谢导师给的建议和指导以及师兄师姐师弟师妹们的鼓励和帮助。

参考文献

[1] Kalman, R.E. (1960) Contributions to the Theory of Optimal Control. *Boletín de la Sociedad Matemática Mexicana*, **5**, 102-119.

[2] Wonham, W.M. (1968) On a Matrix Riccati Equation of Stochastic Control. *SIAM Journal on Control*, **6**, 681-697. https://doi.org/10.1137/0306044

[3] Wonham, W.M. (1967) Optimal Stationary Control of a Linear System with State-Dependent Noise. *SIAM Journal on Control*, **5**, 486-500. https://doi.org/10.1137/0305028

[4] Bismut, J.-M. (1976) Linear Quadratic Optimal Stochastic Control with Random Coefficients. *SIAM Journal on Control and Optimization*, **14**, 419-444. https://doi.org/10.1137/0314028

[5] Pronzato, L., Kulcsár, C. and Walter, E. (1996) An Actively Adaptive Control Policy for Linear Models. *IEEE Transactions on Automatic Control*, **41**, 855-858. https://doi.org/10.1109/9.506238

[6] Chen, S., Li, X. and Zhou, X.Y. (1998) Stochastic Linear Quadratic Regulators with Indefinite Control Weight Costs. *SIAM Journal on Control and Optimization*, **36**, 1685-1702. https://doi.org/10.1137/S0363012996310478

[7] Chen, S. and Zhou, X.Y.U. (2000) Stochastic Linear Quadratic Regulators with Indefinite Control Weight Costs. II. *SIAM Journal on Control and Optimization*, **39**, 1065-1081. https://doi.org/10.1137/S0363012998346578

[8] Rami, M.A., Moore, J.B. and Zhou, X.Y. (2002) Indefinite Stochastic Linear Quadratic Control and Generalized Dif-

ferential Riccati Equation. *SIAM Journal on Control and Optimization*, **40**, 1296-1311. https://doi.org/10.1137/S0363012900371083

[9] Wang, T., Zhang, H. and Luo, Y. (2016) Infinite-Time Stochastic Linear Quadratic Optimal Control for Unknown Discrete-Time Systems Using Adaptive Dynamic Programming Approach. *Neurocomputing*, **171**, 379-386. https://doi.org/10.1016/j.neucom.2015.06.053

[10] Du, K., Meng, Q. and Zhang, F. (2022) A Q-Learning Algorithm for Discrete-Time Linear-Quadratic Control with Random Parameters of Unknown Distribution: Convergence and Stabilization. *SIAM Journal on Control and Optimization*, **60**, 1991-2015. https://doi.org/10.1137/20M1379605

[11] Ziebart, B.D., Maas, A.L., Bagnell, J.A., *et al*. (2008) Maximum Entropy Inverse Reinforcement Learning. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, **8**, 1433-1438.

[12] Boularias, A., Kober, J. and Peters, J. (2011) Relative Entropy Inverse Reinforcement Learning. *Proceedings of the* 14*th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, 11-13 April 2011, 182-189.

[13] Haarnoja, T., Zhou, A., Abbeel, P., *et al*. (2018) Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *The* 35*th International Conference on Machine Learning*, Stockholm, 10-15 July 2018, 1861-1870.

[14] Haarnoja, T., Tang, H., Abbeel, P., *et al*. (2017) Reinforcement Learning with Deep Energy-Based Policies. *The* 34*th International Conference on Machine Learning*, Sydney, 6-11 August 2017, 1352-1361.

[15] Zhao, R., Sun, X. and Tresp, V. (2019) Maximum Entropy-Regularized Multi-Goal Reinforcement Learning. *The* 36*th International Conference on Machine Learning*, Long Beach, 10-15 June 2019, 7553-7562.

[16] Wang, H., Zariphopoulou, T. and Zhou, X.Y. (2020) Reinforcement Learning in Continuous Time and Space: A Stochastic Control Approach. *Journal of Machine Learning Research*, **21**, 1-34.

[17] Wang, H. and Zhou, X.Y. (2020) Continuous-Time Mean-Variance Portfolio Selection: A Reinforcement Learning Framework. *Mathematical Finance*, **30**, 1273-1308. https://doi.org/10.1111/mafi.12281

[18] Bertsekas, D. (2019) Reinforcement Learning and Optimal Control. Athena Scientific, Nashua.

[19] Watkins, C.J.C.H. (1989) Learning from Delayed Rewards. King's College London, London.