

修士論文

反復型軌道積分最適制御の近似解法に関する検討

安並 健太郎

2013年3月15日

奈良先端科学技術大学院大学
情報科学研究科 情報科学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士(工学) 授与の要件として提出した修士論文である。

安並 健太郎

審査委員：

杉本 謙二 教授	(主指導教員)
池田 和司 教授	(副指導教員)
平田 健太郎 准教授	(副指導教員)
松原 崇充 助教	(副指導教員)

反復型軌道積分最適制御の近似解法に関する検討*

安並 健太郎

内容梗概

動的システムの挙動が確率的に決定される非線形確率最適制御問題に対する最適制御則の導出方法の一つである軌道積分確率最適制御法 (PI : Path Integral control) では, 最適解を既知関数の期待値の形で効率的に求められるが, 制御対象とコスト関数に特殊な仮定を要するため適用対象が制限される. この制限を取り除くため, 近年, そのクラス拡張手法である反復型軌道積分確率最適制御法 (IPI : Iterative Path Integral control) が提案された. この手法は上述の制限を解除し, より広いクラスの問題を扱えるようにするだけでなく, PI の数値計算上の問題にも対処できる可能性があるが, その計算にはシステムの動作軌道のサンプルが膨大に必要なため, 実用的な近似解法が重要であると考えられる. そこで本研究では, IPI の近似解法として, 局所 LQA (Linear Quadratic Approximator) に基づく近似解法と, 最近傍回帰に基づく近似解法という 2 つの解法を提案し, シミュレーション実験によりその有効性を確認した.

キーワード

非線形確率最適制御, 軌道積分最適制御, 局所線形 2 次近似

*奈良先端科学技術大学院大学 情報科学研究科 情報科学専攻 修士論文, NAIST-IS-MT123456, 2013 年 3 月 15 日.

Approximate Methods of Iterative Path Integral Stochastic Optimal Control*

Kentaro Yasunami

Abstract

Path Integral control (PI) is one of the methods for nonlinear stochastic optimal control. In PI, the optimal control inputs are derived as the expectation value of a known function. Although this method is very useful, some assumptions on dynamics and cost function are required and therefore restrict its application. To solve this problem, an iterative solution for the path integral method (IPI) have been proposed. This method solves the nonlinear stochastic optimal control problem without imposing the given assumptions. However, to be able to solve computation problems of PI, many sample trajectories are needed, that may not be applicable for most practical problems. In this paper, we propose approximate methods of IPI based on the nearest neighbor method, and by Linear Quadratic Approximator (LQA). Finally, the effectiveness of each method is evaluated by using simulations.

Keywords:

nonlinear stochastic optimal control, path integral control, linear quadratic approximation

*Master's Thesis, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT123456, March 15, 2013.

目次

1. はじめに	1
1.1 非線形確率最適制御	1
1.2 研究目的	2
1.3 本論文の構成	3
2. 準備	4
2.1 軌道積分最適制御法	4
2.2 反復型軌道積分最適制御法	9
2.3 IPI の計算上の課題	11
3. 提案法	13
3.1 提案法の構成	13
3.2 提案法 1：最近傍法を用いた近似法	14
3.3 提案法 2：局所線形 2 次近似を用いた近似法	16
3.3.1 局所線形 2 次近似	16
3.3.2 提案法の着想	18
3.4 提案法 1 と 2 の比較	18
4. シミュレーションによる評価	21
4.1 ダブルスリット問題	21
4.1.1 問題設定	21
4.1.2 シミュレーション結果	22
4.1.3 考察	24
4.2 ロボットシミュレータへの適用	30
4.2.1 4 関節バレットシミュレータ	30
4.2.2 シミュレーション結果	31
4.2.3 考察	32
5. おわりに	34

謝辞	35
参考文献	36
付録	38
A. LQG 制御問題の詳細計算	38

図 目 次

1	IPI の計算における問題の模式図	13
2	ダブルスリット問題	21
3	ダブルスリット ($\Sigma_{\epsilon} = 1$)	25
4	ダブルスリット ($\Sigma_{\epsilon} = 4$)	25
5	ダブルスリット ($\Sigma_{\epsilon} = 9$)	26
6	ダブルスリット ($\Sigma_{\epsilon} = 1$) : 確率ノイズなし	26
7	入力 ($\Sigma_{\epsilon} = 1$)	27
8	ダブルスリット ($\Sigma_{\epsilon} = 4$) : 確率ノイズなし	27
9	入力 ($\Sigma_{\epsilon} = 4$)	28
10	ダブルスリット ($\Sigma_{\epsilon} = 9$) : 確率ノイズなし	28
11	入力 ($\Sigma_{\epsilon} = 9$)	29
12	WAM Arm 実機	30
13	WAM Arm シミュレーション	30
14	アーム軌道 (IPI-LQA)	32
15	アーム軌道 (LQA 補正なし)	32
16	関節角度 (IPI-LQA)	32
17	関節角度 (LQA 補正なし)	32
18	関節角速度 (IPI-LQA)	33
19	関節角速度 (LQA 補正なし)	33
20	学習曲線	33

表 目 次

1	各手法の結果比較, $\Sigma_{\epsilon} = 1$, (200 回試行)	23
2	各手法の結果比較, $\Sigma_{\epsilon} = 4$ (他の設定に関しては Table1 と同様)	23
3	各手法の結果比較, $\Sigma_{\epsilon} = 9$ (他の設定に関しては Table1 と同様)	24

1. はじめに

1.1 非線形確率最適制御

近年、災害救助や生活補助など、従来の工場内などの限定された空間に留まらない幅広い分野で活動するロボットの研究開発が盛んに行われ、ロボットが人間を様々な場面でサポートすることが期待されている。その中で、外乱の下でもロボットが目的の作業を達成する技術が注目されている。ロボットに外乱の下でも目的の動作を達成させる手法の一つとして、最適制御の1手法である非線形確率最適制御法がある。

これは、システムの挙動が確率的に決定されるシステムである非線形確率システムに対して、あるコスト関数で表されるような制御目標を達成する最適制御入力を求める方法である。よく知られた方法としては、最適なコストを動的計画法によって再帰的に計算することで最適入力を得るマルコフ決定過程 (MDP : Markov Decision Process) を用いた価値反復による方策決定 [1] がある。また、Todorov らは非線形確率システムに対して局所的な線形近似を行い、修正リカッチ方程式を解くことで最適制御入力を求める iLQG (Iterative Linear Quadratic Gaussian) 制御法 [2] を提案している。これらの手法は反復計算によって値関数を更新していくことで最適制御入力を求める手法である。

一方、非線形確率最適制御問題を解くことは最終的に確率ハミルトン・ヤコビ (SHJ : Stochastic Hamilton Jacobi) 方程式を解く問題に帰着するが [3]、これは非線形な2階偏微分方程式であるため、解析解を求めることが困難であり、その解を求めるために、様々な手法が提案されている。

そのうちの一つとして、Kappen によって軌道積分確率最適制御法 [4] (PI : Path Integral control) という統計物理学の知見を用いたアプローチが提案されている。この手法はまず制御対象と評価関数のクラスに特殊な仮定を設け、その仮定の下で SHJ 方程式にある指数変換を行うことで、SHJ の非線形項を相殺し、線形の偏微分方程式に変形する。この式は確率論の分野で有名なコルモゴロフの後退方程式の形を持つため、対応する公式である Feynman-Kac の公式によって陽に解析解を求めることができ、結果として SHJ 方程式の解が得られる。この解析解

はある既知な関数の期待値の形で与えられるため、モンテカルロサンプリングなどの統計物理学の手法を用いることで効率的に解くことが可能である。また、サンプルを無限にとった場合の真の解への収束も保証されている。また、この方法を基にした強化学習応用として Theodorou ら [5] は PI^2 (Policy Improvement with Path Integral) と呼ばれる手法を提案し、ロボットの強化学習における有効性を示している。

しかし、PI では SHJ の非線形項を消去するために特殊な仮定を置いており、制御対象と評価関数のクラスに制限があるという問題があった。これに対して、近年、佐藤らによってこの PI のクラス制限を排した、反復型軌道積分最適制御法 [6] (IPI : Iterative Path Integral control) が提案されている。この方法は、PI における制御対象とコスト関数への仮定を用いない代わりに、制御対象のダイナミクスとコスト関数を更新していく反復型のアルゴリズムであり、PI に比べより広いクラスの問題に適用が可能であるという利点を持つ。

IPI は PI におけるクラス制約を排した、より広いクラスへ適用可能な非線形確率最適制御法であるが、この方法は現在、理論と最適入力 of 導出方法については提案されているが、実用的な解法については十分には議論されていない。反復計算において必要な値関数を求める際に複雑な入れ子構造の計算を解かねばならず、膨大なサンプルが必要となるため、実用性に乏しく、特に高次元システムへの適用は困難である。

1.2 研究目的

本研究では IPI を効率的に解くために、IPI の値関数を計算する際の入れ子構造を関数近似によって回避する近似解法を検討する。これによって計算に必要なサンプル数を抑えた実用的な解法を目指す。近似手法として、最近傍法と局所 LQA (Linear Quadratic Approximator) という 2 種類の関数近似を用いた近似解法を提案し、その有効性についてシミュレーションを用いて確認する。

1.3 本論文の構成

本論文の構成について述べる.

本論文では, 2 章で PI と IPI の詳細な導出過程の説明とその数値計算において発生する問題について説明し, 3 章でその数値計算上の問題を解決するための近似解法として, 最近傍法を用いた近似解法と, 局所 LQA を用いた近似解法の 2 種類の提案法について説明する. 4 章でダブルスリット問題とロボットアームのシミュレーションによって提案法の有効性を検証し, 5 章で提案法のまとめと今後の課題について述べる.

2. 準備

本章では提案手法の背景となる軌道積分最適制御法および反復型軌道積分最適制御法の導出過程と，反復型軌道積分最適制御法の計算上の問題に関して述べる．

2.1 軌道積分最適制御法

システムの状態 $\mathbf{x}_t \in \mathbb{R}^n$ ，入力 $\mathbf{u}_t \in \mathbb{R}^m$ として，以下のような非線形確率システムを考える．

$$\dot{\mathbf{x}}_t = \mathbf{f}(\mathbf{x}_t, t) + \mathbf{g}(\mathbf{x}_t, t)\mathbf{u}_t + \mathbf{h}(\mathbf{x}_t, t)\epsilon_t \quad (1)$$

ここで， ϵ_t は平均 0，共分散行列 Σ_ϵ のガウスノイズである．

本論文では時間区間 $t \in [0, t_N]$ での制御問題を考えることとする．このシステムに対して，以下のようなコスト関数を考える．

$$R(t_i \rightarrow t_N) = \phi_{t_N} + \int_{t_i}^{t_N} r(\mathbf{x}_t, \mathbf{u}_t, t) dt \quad (2)$$

ここで $R(t_i \rightarrow t_N)$ は時刻 t_i から t_N までの軌道コスト， ϕ_{t_N} は終端コストである．

また，逐次コスト $r(\mathbf{x}_t, \mathbf{u}_t, t)$ を以下のように設定する．

$$r(\mathbf{x}_t, \mathbf{u}_t, t) = q(\mathbf{x}_t, t) + \frac{1}{2} \mathbf{u}_t^T \mathbf{R} \mathbf{u}_t \quad (3)$$

ここで，値関数 V_{t_i} を以下のように定義する．

$$V(\mathbf{x}_{t_i}, t_i) = V_{t_i} = \min_{\mathbf{u}_{t_i \rightarrow t_N}} \mathbb{E}[R(t_i \rightarrow t_N)] \quad (4)$$

値関数は最適な入力系列が加えられたときのコストの値を表す関数であり， $\mathbb{E}$ は，状態 $\mathbf{x}_{t_i}$ からスタートするすべての軌道に対する期待値である．以下では表記の簡単化のため，逐次コストを $r_t = r(\mathbf{x}_t, \mathbf{u}_t, t)$ ， $q_t = q(\mathbf{x}_t, t)$ ，ダイナミクスを $\mathbf{F}(\mathbf{x}_t, \mathbf{u}, t) = \mathbf{f}(\mathbf{x}_t, t) + \mathbf{g}(\mathbf{x}_t, t)\mathbf{u}$ ，アクティブダイナミクスとパッシブダイナミクスを $\mathbf{f}_t = \mathbf{f}(\mathbf{x}_t, t)$ ， $\mathbf{g}_t = \mathbf{g}(\mathbf{x}_t, t)$ ， $\mathbf{h}_t = \mathbf{h}(\mathbf{x}_t, t)$ とおく．

値関数が上記のように与えられた場合、 V_t が満たすべき方程式である確率ハミルトン・ヤコビ・ベルマン (SHJB : Stochastic Hamilton Jacobi Bellman) 方程式は次式のように与えられる.

$$-\frac{\partial V_t}{\partial t} = \min_{\mathbf{u}_t} \left(r_t + (\nabla_{\mathbf{x}} V_t)^T \mathbf{F}(\mathbf{x}_t, \mathbf{u}_t, t) + \frac{1}{2} \text{tr} \left((\nabla_{\mathbf{xx}} V_t) \mathbf{h}_t \Sigma_{\epsilon} \mathbf{h}_t^T \right) \right) \quad (5)$$

この式を $\mathbf{u}_t$ に関して偏微分し、0 と置くことで対象とするシステムに対する最適制御入力 that 以下のように得られる.

$$\mathbf{u}_t = -\mathbf{R}^{-1} \mathbf{g}_t^T (\nabla_{\mathbf{x}_t} V_t) \quad (6)$$

この最適入力を (5) 式に代入して入力項を消去すると、次式のような確率ハミルトンヤコビ方程式が得られる.

$$-\frac{\partial V_t}{\partial t} = q_t + (\nabla_{\mathbf{x}} V_t)^T \mathbf{f}_t - \frac{1}{2} (\nabla_{\mathbf{x}} V_t)^T \mathbf{g}_t \mathbf{R}^{-1} \mathbf{g}_t^T (\nabla_{\mathbf{x}} V_t) + \frac{1}{2} \text{tr} \left((\nabla_{\mathbf{xx}} V_t) \mathbf{h}_t \Sigma_{\epsilon} \mathbf{h}_t^T \right) \quad (7)$$

$\nabla_{\mathbf{x}}$, $\nabla_{\mathbf{xx}}$ はそれぞれヤコビアン, ヘシアンを表している.

対象とする確率最適制御問題を解くことは、この SHJ 方程式を解くことと等価であり、この SHJ 方程式の解が求まれば (6) 式を用いて最適制御入力は陽に求めることができる.

いま、値関数 V_t に関して、次式のような指数変換

$$\psi_t = \exp -\frac{1}{\lambda} V_t \quad (8)$$

を施したものを定義する. このとき、値関数の時間、状態に対する偏微分は以下のように表せる.

$$\frac{\partial V_t}{\partial t} = -\lambda \frac{1}{\psi_t} \frac{\partial \psi_t}{\partial t}, \quad (9)$$

$$\nabla_{\mathbf{x}} V_t = -\lambda \frac{1}{\psi_t} \nabla_{\mathbf{x}} \psi_t, \quad (10)$$

$$\nabla_{\mathbf{xx}} V_t = \lambda \frac{1}{\psi_t^2} \nabla_{\mathbf{x}} \psi_t (\nabla_{\mathbf{x}} \psi_t)^T - \lambda \frac{1}{\psi_t} \nabla_{\mathbf{xx}} \psi_t. \quad (11)$$

これを (7) 式に代入して整理することで、次式を得る.

$$\begin{aligned} \frac{\partial \psi_t}{\partial t} - \frac{q_t}{\lambda} \psi_t + (\nabla_{\mathbf{x}} \psi_t)^T \mathbf{f}_t + \frac{\lambda}{2\psi_t} (\nabla_{\mathbf{x}} \psi_t)^T \mathbf{g}_t \mathbf{R}^{-1} \mathbf{g}_t^T (\nabla_{\mathbf{x}} \psi_t) \\ - \frac{\lambda}{2\psi_t} (\nabla_{\mathbf{x}} \psi_t)^T \mathbf{h}_t \frac{\Sigma_{\epsilon}}{\lambda} \mathbf{h}_t^T (\nabla_{\mathbf{x}} \psi_t) + \frac{1}{2} \text{tr} ((\nabla_{\mathbf{xx}} \psi_t) \mathbf{h}_t \Sigma_{\epsilon} \mathbf{h}_t^T) = 0, \\ \psi_{t_N} = \exp \left(-\frac{1}{\lambda} \phi_{t_N} \right) \end{aligned} \quad (12)$$

ここで、対象とするシステムに対して、以下の二つの仮定

1. 全ての時刻 t において、 $\mathbf{g}(\mathbf{x}_t, t) = \mathbf{h}(\mathbf{x}_t, t)$.
2. コストの入力重み $\mathbf{R}$ と制御対象のノイズ共分散行列 $\mathbf{Q}$ の間に、 $\lambda \mathbf{R}^{-1} = \Sigma_{\epsilon}$ が成り立つ.

を置くと、 $\mathbf{g}_t \mathbf{R}^{-1} \mathbf{g}_t^T = \mathbf{h}_t \frac{\Sigma_{\epsilon}}{\lambda} \mathbf{h}_t^T$ となるので、(12) 式の左辺第 4 項と第 5 項の非線形項が相殺され、次式のような線形 2 階偏微分方程式の形に整理することができる.

$$\begin{aligned} \frac{\partial \psi_t}{\partial t} &= \frac{q_t}{\lambda} \psi_t - (\nabla_{\mathbf{x}} \psi_t)^T \mathbf{f}_t - \frac{1}{2} \text{tr} ((\nabla_{\mathbf{xx}} \psi_t) \mathbf{g}_t \Sigma_{\epsilon} \mathbf{g}_t^T), \\ \psi_{t_N} &= \exp \left(-\frac{1}{\lambda} \phi_{t_N} \right) \end{aligned} \quad (13)$$

この偏微分方程式は、確率論におけるコルモゴロフの後退方程式の構造を持つ.

コルモゴロフの後退方程式には、Feynman-Kac の公式と呼ばれる解法が存在し [7], その解析解は次式のような軌道に対するコストの期待値の形で得られる.

$$\psi_{t_i} = \mathbb{E}_{\tau_i} \left[\exp \left(-\frac{1}{\lambda} \phi_{t_N} - \int_{t_i}^{t_N} \frac{q_t}{\lambda} dt \right) \right] \quad (14)$$

この期待値計算はタイムステップ $dt = t_{i+1} - t_i$ と $\mathbf{x}$ の時系列 $\tau_i = \{\mathbf{x}_{t_i}, \mathbf{x}_{t_{i+1}}, \dots, \mathbf{x}_{t_N}\}$ を考えると、統計物理学の分野で知られている軌道積分の考え方をを用いて次式のように計算することができる.

$$\psi_{t_i} = \lim_{dt \rightarrow 0} \int p(\tau_i | \mathbf{x}_{t_i}) \exp \left[-\frac{1}{\lambda} \left(\phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} dt \right) \right] d\tau_i \quad (15)$$

いま $p(\tau_i|\mathbf{x}_{t_i})$ はある時刻 t_i での状態 $\mathbf{x}_{t_i}$ に対する、時系列 τ_i の生起確率であり、

$$\begin{aligned} p(\tau_i|\mathbf{x}_{t_i}) &= p(\mathbf{x}_{t_N}, \dots, \mathbf{x}_{t_{i+1}}|\mathbf{x}_{t_i}) \\ &= \prod_{j=i}^{N-1} p(\mathbf{x}_{t_{j+1}}|\mathbf{x}_{t_j}) \end{aligned} \quad (16)$$

ように展開できる．結局，軌道の生起確率はガウス分布の式から，次式によって求めることができる．

$$\begin{aligned} p(\tau_i|\mathbf{x}_{t_i}) &= \frac{\exp\left(-\frac{1}{2} \sum_{j=1}^{N-1} \|\mathbf{x}_{t_{j+1}} - \mathbf{x}_{t_j} - \mathbf{f}_{t_j} dt\|_{\Sigma_{t_j}^{-1}}^2\right)}{\prod_{j=i}^{N-1} ((2\pi)^l |\Sigma_{t_j}|)^{1/2}} \\ &= \frac{\exp\left(-\frac{1}{2\lambda} \sum_{j=1}^{N-1} \left\| \frac{\mathbf{x}_{t_{j+1}} - \mathbf{x}_{t_j}}{dt} - \mathbf{f}_{t_j} \right\|_{\mathbf{H}_{t_j}^{-1}}^2 dt\right)}{\prod_{j=i}^{N-1} ((2\pi)^l |\Sigma_{t_j}|)^{1/2}} \end{aligned} \quad (17)$$

ここで， $\Sigma_t = \mathbf{h}_t \Sigma_\epsilon \mathbf{h}_t^T dt = \lambda \mathbf{g}_t \mathbf{R}^{-1} \mathbf{g}_t^T dt$ ， $\mathbf{H}_t = \mathbf{g}_t \mathbf{R}^{-1} \mathbf{g}_t^T$ である．

(17) 式を (15) 式に代入することで以下の式を得る．

$$\begin{aligned} \psi_{t_i} &= \lim_{dt \rightarrow 0} \int \frac{\exp\left(-\frac{1}{2\lambda} \sum_{j=1}^{N-1} \left\| \frac{\mathbf{x}_{t_{j+1}} - \mathbf{x}_{t_j}}{dt} - \mathbf{f}_{t_j} \right\|_{\mathbf{H}_{t_j}^{-1}}^2 dt\right)}{\prod_{j=i}^{N-1} ((2\pi)^l |\Sigma_{t_j}|)^{1/2}} \exp\left[-\frac{1}{\lambda} \left(\phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} dt\right)\right] d\tau_i \\ &= \lim_{dt \rightarrow 0} \int \frac{\exp\left(-\frac{1}{2\lambda} \sum_{j=1}^{N-1} \left\| \frac{\mathbf{x}_{t_{j+1}} - \mathbf{x}_{t_j}}{dt} - \mathbf{f}_{t_j} \right\|_{\mathbf{H}_{t_j}^{-1}}^2 dt - \frac{1}{\lambda} \left(\phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} dt\right)\right)}{\prod_{j=i}^{N-1} ((2\pi)^l |\Sigma_{t_j}|)^{1/2}} d\tau_i \\ &= \lim_{dt \rightarrow 0} \int \frac{\exp\left(-\frac{1}{\lambda} S(\tau_i)\right)}{\prod_{j=i}^{N-1} ((2\pi)^l |\Sigma_{t_j}|)^{1/2}} d\tau_i \end{aligned} \quad (18)$$

$S(\tau_i)$ は次式で表される軌道コストである．

$$\begin{aligned} S(\tau_i) &= \phi_{t_N} + \sum_{j=i}^{N-1} q(\mathbf{x}_{t_j}, t_j) dt + \frac{1}{2} \sum_{j=i}^{N-1} \left\| \frac{\mathbf{x}_{t_{j+1}} - \mathbf{x}_{t_j}}{dt} - \mathbf{f}_{t_j} \right\|_{\mathbf{H}_{t_j}^{-1}}^2 dt \\ &= \phi_{t_N} + \sum_{j=i}^{N-1} q(\mathbf{x}_{t_j}, t_j) dt + \frac{1}{2} \sum_{j=i}^{N-1} \left\| \mathbf{g}_{t_j} \mathbf{u}_{t_j} + \mathbf{h}_{t_j} \epsilon_{t_j} \right\|_{\mathbf{H}_{t_j}^{-1}}^2 dt \end{aligned} \quad (19)$$

したがって ψ_{t_i} の $\mathbf{x}$ に関する偏微分は

$$\begin{aligned}
\nabla_{\mathbf{x}_{t_i}} \psi_{t_i} &= \lim_{dt \rightarrow 0} \nabla_{\mathbf{x}_{t_i}} \int \frac{\exp\left(-\frac{1}{\lambda} S(\tau_i)\right)}{\prod_{j=i}^{N-1} ((2\pi)^l |\boldsymbol{\Sigma}_{t_j}|)^{1/2}} d\tau_i \\
&\propto \lim_{dt \rightarrow 0} \nabla_{\mathbf{x}_{t_i}} \int \exp\left(-\frac{1}{\lambda} S(\tau_i)\right) d\tau_i \\
&= \lim_{dt \rightarrow 0} \int \nabla_{\mathbf{x}_{t_i}} \exp\left(-\frac{1}{\lambda} S(\tau_i)\right) d\tau_i \\
&= \lim_{dt \rightarrow 0} \int \exp\left(-\frac{1}{\lambda} S(\tau_i)\right) \nabla_{\mathbf{x}_{t_i}} \left(-\frac{1}{\lambda} S(\tau_i)\right) d\tau_i \\
&= -\frac{1}{\lambda} \int \exp\left(-\frac{1}{\lambda} S(\tau_i)\right) \lim_{dt \rightarrow 0} (\nabla_{\mathbf{x}_{t_i}} S(\tau_i)) d\tau_i \\
&= \frac{1}{\lambda} (\mathbf{g}_{t_i} \mathbf{R}^{-1} \mathbf{g}_{t_i}^T)^{-1} \mathbf{g}_{t_i} \int \exp\left(-\frac{1}{\lambda} S(\tau_i)\right) d\tau_i
\end{aligned} \tag{20}$$

となる． よって次式を得る．

$$\begin{aligned}
\nabla_{\mathbf{x}_{t_i}} \log \psi_{t_i} &= \frac{\nabla_{\mathbf{x}_{t_i}} \psi_{t_i}}{\psi_{t_i}} \\
&= \frac{1}{\lambda} (\mathbf{g}_{t_i} \mathbf{R}^{-1} \mathbf{g}_{t_i}^T)^{-1} \mathbf{g}_{t_i} \int \frac{\exp -\frac{1}{\lambda} S(\tau_i)}{\int \exp -\frac{1}{\lambda} S(\tau_i) d\tau_i} \epsilon_{t_i} d\tau_i
\end{aligned} \tag{21}$$

(6) 式に (21) 式を代入することで， 次式のように最適入力を求めることができる．

$$\begin{aligned}
\mathbf{u}_{t_i} &= \lambda \mathbf{R}^{-1} \mathbf{g}_{t_i}^T \nabla_{\mathbf{x}} \log \psi_{t_i} \\
&= \int \frac{\exp -\frac{1}{\lambda} S(\tau_i)}{\int \exp -\frac{1}{\lambda} S(\tau_i) d\tau_i} \epsilon_{t_i} d\tau_i \\
&= \int P(\tau_i) \epsilon_{t_i} d\tau_i
\end{aligned} \tag{22}$$

ここで， $P(\tau_i) = \frac{\exp -\frac{1}{\lambda} S(\tau_i)}{\int \exp -\frac{1}{\lambda} S(\tau_i) d\tau_i}$ である．

この式から， 非線形確率システムの最適制御入力が， ある確率測度上でのシステムのノイズの平均で表されることが分かる． これは従来の非線形確率最適制御では見られない PI の特徴の一つである．

また、これらの値はモンテカルロサンプリングによって以下のように計算することができる。

k 本目のサンプル軌道を τ_i^k とおくと、時刻 t_i における ψ_{t_i} と最適入力 $\mathbf{u}_{t_i}$ は、 K 本のサンプル軌道を用いて、以下のように近似計算される。

$$\psi_{t_i} = \frac{\sum_{k=1}^K \exp\left(-\frac{1}{\lambda} S(\tau_i^k)\right)}{K} \quad (23)$$

$$\nabla_{\mathbf{x}_{t_i}} \log \psi_{t_i} = \frac{1}{\lambda} (\mathbf{g}_{t_i} \mathbf{R}^{-1} \mathbf{g}_{t_i}^T)^{-1} \mathbf{g}_{t_i} \sum_{k=1}^K \frac{\exp\left(-\frac{1}{\lambda} S(\tau_i^k)\right)}{\sum_{k=1}^K \exp\left(-\frac{1}{\lambda} S(\tau_i^k)\right)} \epsilon_{t_i} \quad (24)$$

$$\begin{aligned} \mathbf{u}_{t_i} &= \int P(\tau_i) \epsilon_{t_i} d\tau_i \\ &= \sum_{k=1}^K \frac{\exp\left(-\frac{1}{\lambda} S(\tau_i^k)\right)}{\sum_{k=1}^K \exp\left(-\frac{1}{\lambda} S(\tau_i^k)\right)} \epsilon_{t_i} \end{aligned} \quad (25)$$

モンテカルロサンプリングの性質より、これらの値は $K \rightarrow \infty$ で真値へ収束することが保証される。

2.2 反復型軌道積分最適制御法

前述した PI では、SHJ 方程式に対して仮定 1,2 を置く事で、2 次の項を打ち消して SHJ 方程式をコルモゴロフの後退方程式の形に変形していたが、この仮定 1,2 によって適用可能なクラスが制限されるとともに、 λ の値次第でコストに対する計算結果の感度が極端に大きくなったり小さくなったりして数値計算が不安定化する問題がある。Theodorou ら [5] は経験的な λ を用いて PI を解くことでこの数値計算上の問題を回避する方法を提案しているが、この方法は PI の 2 つの仮定を満たしておらず、最適性を満たす理論的な保証がない。これらの問題に対して有効な手段として、佐藤らが提案した IPI[6] がある。この手法は PI における仮定 1,2 を排した PI のクラス拡張手法である。IPI では PI の仮定 1,2 の代わりに、SHJ の 2 次の項を適当な関数とおいて PI を近似的に解き、それによって得られた値関数を 2 次の項に代入する反復計算を行うことで真の解を求める。以下

ではその具体的な手法について説明する.

(12) 式を基に次式のような変形を行う.

$$\begin{aligned} \frac{\partial \psi_t}{\partial t} - \frac{q_t}{\lambda} \psi_t + (\nabla_{\mathbf{x}} \psi_t)^T \mathbf{f}_t + \frac{\lambda}{2\psi_t} (\nabla_{\mathbf{x}} \psi_t)^T \mathbf{g}_t \mathbf{R}^{-1} \mathbf{g}_t^T (\nabla_{\mathbf{x}} \psi_t) \\ - \frac{\lambda}{2\psi_t} (\nabla_{\mathbf{x}} \psi_t)^T \mathbf{h}_t \frac{\Sigma_{\epsilon}}{\lambda} \mathbf{h}_t^T (\nabla_{\mathbf{x}} \psi_t) + \frac{1}{2} \text{tr} \left((\nabla_{\mathbf{xx}} \psi_t) \mathbf{h}_t \Sigma_{\epsilon} \mathbf{h}_t^T \right) \end{aligned} \quad (26)$$

これを基に、以下のような関数を定義する.

$$\begin{aligned} H(\psi, \eta) := \frac{\partial \psi_t}{\partial t} - \frac{1}{\lambda} \left(q_t + \frac{\lambda^2}{2} (\nabla_{\mathbf{x}} \log \eta)^T \nu_{t_i} (\nabla_{\mathbf{x}} \log \eta) \right) \psi_t \\ + (\nabla_{\mathbf{x}} \psi_t)^T \left(\mathbf{f}_t + \lambda \nu_{t_i} (\nabla_{\mathbf{x}} \log \eta) \right) + \frac{1}{2} \text{tr} \left((\nabla_{\mathbf{xx}} \psi_t) \mathbf{h}_t \Sigma_{\epsilon} \mathbf{h}_t^T \right) \end{aligned} \quad (27)$$

ここで $\nu_t = \mathbf{g}_t \mathbf{R}^{-1} \mathbf{g}_t^T - \frac{1}{\lambda} \mathbf{h}_t \Sigma_{\epsilon} \mathbf{h}_t^T$ である. いま, 逐次コストとパッシブダイナミクスに対して

$$\hat{q}_t = q_t + \frac{\lambda^2}{2} (\nabla_{\mathbf{x}} \log \eta)^T \nu_{t_i} (\nabla_{\mathbf{x}} \log \eta) \quad (28)$$

$$\hat{\mathbf{f}}_t = \mathbf{f}_t + \lambda \nu_{t_i} (\nabla_{\mathbf{x}} \log \eta) \quad (29)$$

と置くと, (27) 式は

$$H(\psi, \eta) = \frac{\partial \psi_t}{\partial t} - \frac{\hat{q}_t}{\lambda} \psi_t + (\nabla_{\mathbf{x}} \psi_t)^T \hat{\mathbf{f}}_t + \frac{1}{2} \text{tr} \left((\nabla_{\mathbf{xx}} \psi_t) \mathbf{h}_t \Sigma_{\epsilon} \mathbf{h}_t^T \right) \quad (30)$$

となり、前述したコルモゴロフの後退方程式の形をとるため、Feynman-Kac の公式が適用でき、PI によって解を求めることができる.

この発想を用いて、 $H(\psi_t^{(j)}, \psi_t^{(j-1)}) = 0$ となる $\psi_t^{(j)}$ を求め、反復計算によって最適入力を更新していく解法が考えられる. この時、逐次コストとパッシブダイナミクスが以下のように更新される.

$$q^{(j)}(\mathbf{x}_t, t) := q(\mathbf{x}_t, t) + \frac{\lambda^2}{2} (\nabla_{\mathbf{x}} \log \psi_t^{(j-1)})^T \nu_{t_i} (\nabla_{\mathbf{x}} \log \psi_t^{(j-1)}) \quad (31)$$

$$\mathbf{f}_t^{(j)} := \mathbf{f}_t + \lambda \nu_{t_i} (\nabla_{\mathbf{x}} \log \psi_t^{(j-1)}) \quad (32)$$

ここで

$$q^{(0)}(\mathbf{x}_t, t) := q_t$$

$$\mathbf{f}_t^{(0)} := \mathbf{f}_t$$

であり, $\nabla_{\mathbf{x}} \log \psi_t^{(j-1)} = \nabla_{\mathbf{x}} \log \psi_t^{(j-1)}(\mathbf{x}_t, t)|_{\mathbf{x}_t^{(j)}}$ と定める. これによって j 回目の更新における軌道コスト $S^{(j)}(\tau_i)$, 準最適解 $\psi_{t_i}^{(j)}$ と準最適入力 $\mathbf{u}_{t_i}^{(j)}$ は次式で与えられる.

$$S^{(j)}(\tau_i) = \phi_{t_N} + \sum_{k=i}^{N-1} q^{(j)}(\mathbf{x}_{t_k}, t_k) dt + \frac{1}{2} \sum_{k=i}^{N-1} \|\mathbf{g}_{t_k} \mathbf{u}_{t_k} + \mathbf{h}_{t_k} \epsilon_{t_k}\|_{\mathbf{H}_{t_k}^{-1}}^2 dt \quad (33)$$

$$\psi_{t_i}^{(j)} = \frac{\sum_{k=1}^K \exp\left(-\frac{1}{\lambda} S^{(j)}(\tau_{t_i}^k)\right)}{K} \quad (34)$$

$$\nabla_{\mathbf{x}} \log \psi_{t_i}^{(j)} = \frac{1}{\lambda} (\mathbf{h}_t \frac{\Sigma_{\epsilon}}{\lambda} \mathbf{h}_t^T)^{-1} \mathbf{h}_t \frac{\sum_{k=1}^K \exp\left(-\frac{1}{\lambda} S^{(j)}(\tau_t^k)\right) \epsilon_{t_i, k}^{(j)}}{\sum_{k=1}^K \exp\left(-\frac{1}{\lambda} S^{(j)}(\tau_t^k)\right)} \quad (35)$$

$$\begin{aligned} \mathbf{u}_{t_i}^{(j)} &= \lambda \mathbf{R}^{-1} \mathbf{g}_{t_i}^T \nabla_{\mathbf{x}} \log \psi_{t_i}^{(j)} \\ &= \mathbf{R}^{-1} \mathbf{g}_t^T \left(\mathbf{h}_t \frac{\Sigma_{\epsilon}}{\lambda} \mathbf{h}_t^T \right)^{-1} \mathbf{h}_t \frac{\sum_{k=1}^K \exp\left(-\frac{1}{\lambda} S^{(j)}(\tau_t^k)\right) \epsilon_{t_i, k}^{(j)}}{\sum_{k=1}^K \exp\left(-\frac{1}{\lambda} S^{(j)}(\tau_t^k)\right)} \end{aligned} \quad (36)$$

佐藤らは解の近似誤差を考慮したアルゴリズムの収束性に関しても議論しており [6], この入力と値関数は無限回繰り返しで真の最適解に収束することが示されている.

2.3 IPI の計算上の課題

以下では, IPI の具体的な解法について考える. 佐藤らは IPI の理論提案を行っているが, 実計算の手順については明示していない¹ため, IPI の計算のためにま

¹http://videlectures.net/cyberstat2012_satoh_optimal_control/ において低次元システムでのシミュレーション結果が示されているが, ここでも具体的な数値計算法に関しての明言はされていない

ず，前章で求めた IPI における更新式 (31)，(32) の実計算について考える必要がある．

(21) 式より， j 回目の更新において，補正項内の関数 $\nabla_{\mathbf{x}} \log \psi_{t_i}^{(j-1)}$ は $j-1$ 回目の更新時の条件設定 $\mathbf{f}^{(j-1)}, q^{(j-1)}$ の下でのサンプルから生成される関数であることが分かる．しかし，PI の計算によって得られる $\nabla_{\mathbf{x}} \log \psi_{t_i}^{(j-1)}$ はある最適軌道上の値のみであり，状態空間上の各状態 $\mathbf{x}$ に対する関数としては得られない．また，(31)，(32) 式より $\mathbf{f}^{(j-1)}, q^{(j-1)}$ は $\psi_{t_i}^{(j-2)}$ によって決定されるため，結局 $\psi_{t_i}^{(j-1)}$ の計算は図 1 に示すように， $j-1$ 回の各更新の計算結果を使用する入れ子構造になっている．つまり，任意の $\mathbf{x}_{t_i}$ に対して $\psi_{t_i}^{(j-1)}$ を求めるには $j-1$ 回の各反復計算それぞれで状態空間全体の価値関数を求める必要がある．

しかし，1 回の PI を解いた場合に得られる価値関数は最適軌道上の値のみであるため，厳密解を得るには各時刻毎に PI を解くことになる．つまり，(タイムステップ N) $\times$ (反復回数 $j-1$) 回の PI を解く必要があり，これらすべてをモンテカルロサンプリングに基づく演算で求めるのは計算に必要なサンプル数の観点から現実的ではない．

そこで， $j-1$ 回目の PI によって得られる軌道コストの時系列データを用いて，全状態空間上の価値関数を近似的に求めることを考える．

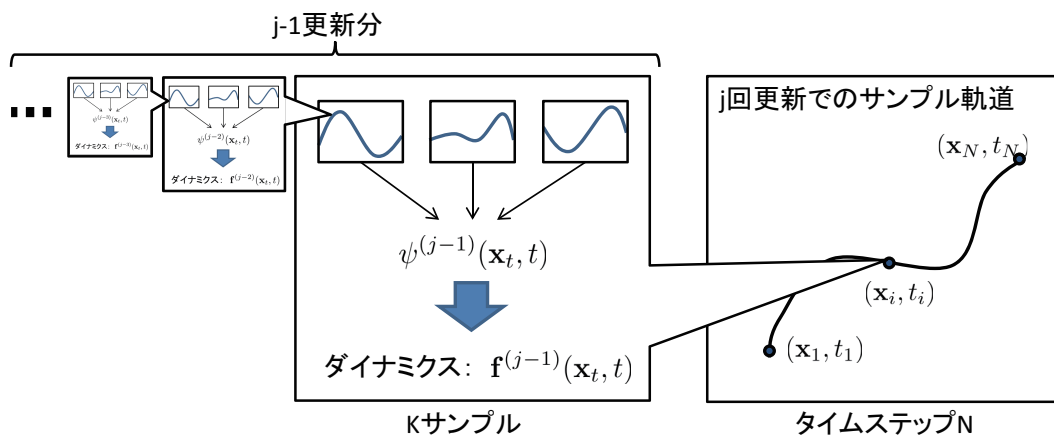


図 1 IPI の計算における問題の模式図

3. 提案法

本章では前章の内容を踏まえ、計算効率の良い近似解法について議論する。

3.1 提案法の構成

前章で、IPI を計算するには各反復毎に値関数を状態空間全体で求める必要があるが、実際の PI の計算ではある初期状態、ダイナミクスとコスト関数に対して 1 つの最適軌道上の値関数値しか求めることができないため、この 1 最適軌道上の値関数値を用いて全状態空間上の値関数を近似する必要があることを述べた。

本章ではこれに関して 2 つの関数近似法を提案する。

3.2 節では最近傍法を用いた関数近似法について述べる。これは、同一時刻で状態が近い場合、値関数の値も近いという仮定の下、複数の初期状態から PI を解き、それらの最適軌道上の値関数を用いて、新しく得られた状態に対応する値関数値を最近傍法によって探索する方法である。

3.3 節では局所線形 2 次近似 (LQA : Linear Quadratic Approximation) を用いた関数近似法について述べる。これは、局所線形ダイナミクスと 2 次コストに対

する値関数が効率的に求められることを利用して、前回の最適軌道を中心としてその周辺状態に対する局所的な値関数の近似を行う解法である。

以下ではこれらの手法の詳しい内容について述べる。

3.2 提案法 1 : 最近傍法を用いた近似法

本節では直観的な近似解法として、最近傍回帰 (nearest neighbor method)[8] を用いた方法について説明する。この方法では、各時刻 t_i において状態 $\mathbf{x}_{t_i}$ の値が近ければ値関数も近い値を示すという仮定の下で、状態をクエリとした最近傍回帰による関数近似を行う。

手順としては、まず $j-1$ 回目の繰り返し計算時に初期状態 $\mathbf{x}_0$ にノイズを加えた M 個の新たな初期状態 $\mathbf{x}_{t_0,m}$ ($1 \leq m \leq M$) を生成し、これらの初期状態に対して PI を解いた解とその時の最適軌道上の状態 $\bar{\mathbf{x}}_{t,m}$ を保存する。その後、 j 回目の繰り返し計算において

$$\hat{m} = \arg \min_m \|\bar{\mathbf{x}}_{t,m}^{(j-1)} - \mathbf{x}_t^{(j)}\| \quad (37)$$

となる $\hat{m}$ を選び、 j 回目の更新時の値を、 $j-1$ 回目の結果を用いて

$$\nabla_{\mathbf{x}} \log \psi^{(j-1)}(\mathbf{x}_t^{(j)}, t) = \nabla_{\mathbf{x}} \log \psi^{(j-1)}(\bar{\mathbf{x}}_{t,\hat{m}}^{(j-1)}, t) \quad (38)$$

と近似する。これを用いて

$$\begin{aligned} q^{(j)}(\mathbf{x}_t, t) &:= q(\mathbf{x}_t, t) \\ &\quad + \frac{\lambda^2}{2} \left(\nabla_{\mathbf{x}} \log \psi^{(j-1)}(\bar{\mathbf{x}}_{t,\hat{m}}^{(j-1)}, t) \right)^T \nu_t \left(\nabla_{\mathbf{x}} \log \psi^{(j-1)}(\bar{\mathbf{x}}_{t,\hat{m}}^{(j-1)}, t) \right) \end{aligned} \quad (39)$$

$$\mathbf{f}_t^{(j)} := \mathbf{f}_t + \lambda \nu_{t_i}(\nabla_{\mathbf{x}} \log \psi^{(j-1)}(\bar{\mathbf{x}}_{t,\hat{m}}^{(j-1)}, t)) \quad (40)$$

とすることで IPI を計算する。具体的な手順について Algorithm 1 に示す。

Algorithm 1 IPI-NN

Input: $\{\epsilon_{t_1}, \epsilon_{t_2}, \dots, \epsilon_{t_N}\}$

Output: $\{\mathbf{u}_{t_1}, \mathbf{u}_{t_2}, \dots, \mathbf{u}_{t_N}\}$

Given: $\frac{\partial \log \psi^{(0)}(\mathbf{x}, t)}{\partial \mathbf{x}}, \mathbf{f}_t^{(0)}, q_t^{(0)}, \mathbf{x}_0$

for $j = 1$ **to** J **do**

- make M random start state $\mathbf{x}_0^{(m)} = \mathbf{x}_0 + (\text{random noise})$

for $m = 1$ **to** M **do**

- $\mathbf{f}_t^{(j)} \Leftarrow \mathbf{f}_t^{(0)} + \lambda \nu_t \frac{\partial \log \psi^{(j-1)}(\mathbf{x}, t)}{\partial \mathbf{x}}$

- $q_t^{(j)} \Leftarrow q_t^{(0)} + \frac{\lambda^2}{2} \frac{\partial \log \psi^{(j-1)}(\mathbf{x}, t)}{\partial \mathbf{x}}^T \nu_t \frac{\partial \log \psi^{(j-1)}(\mathbf{x}, t)}{\partial \mathbf{x}}$

- create K roll-outs of the system from same state $\mathbf{x}_0^{(m)}$

for $k = 1$ **to** K **do**

for $i = 1$ **to** N **do**

- $S(\tau_{i,k}) = \phi_{t_N,k} + \sum_{n=i}^N q_{t_n,k}^{(j)}$

- $P(\tau_{i,k}) = \frac{e^{-\frac{1}{\lambda} S(\tau_{i,k})}}{\sum_{k=1}^K e^{-\frac{1}{\lambda} S(\tau_{i,k})}}$

end for

end for

for $i = 1$ **to** N **do**

- $\frac{\partial \log \psi_{t,m}^{(j)}}{\partial \mathbf{x}} = \lambda \left(\mathbf{h}_t \frac{\Sigma_\epsilon}{\lambda} \mathbf{h}_t^T \right)^{-1} \mathbf{h}_t \sum_{k=1}^K P(\tau_{i,k}) \epsilon_{t_i,k}^{(j)}$

- save datasets $\left(\frac{\partial \log \psi_{t,m}^{(j)}}{\partial \mathbf{x}}, \mathbf{x}_t^{(m)} \right)$

end for

end for

- create K roll-outs of the system from same state $\mathbf{x}_0$

for $k = 1$ **to** K **do**

for $i = 1$ **to** N **do**

- $S(\tau_{i,k}) = \phi_{t_N,k} + \sum_{n=i}^N q_{t_n,k}^{(j)}$

- $P(\tau_{i,k}) = \frac{e^{-\frac{1}{\lambda} S(\tau_{i,k})}}{\sum_{k=1}^K e^{-\frac{1}{\lambda} S(\tau_{i,k})}}$

end for

end for

for $i = 1$ **to** N **do**

- $\mathbf{u}_{t_i} = \mathbf{R}^{-1} \mathbf{g}_t^T \left(\mathbf{h}_t \frac{\Sigma_\epsilon}{\lambda} \mathbf{h}_t^T \right)^{-1} \mathbf{h}_t \sum_{k=1}^K P(\tau_{i,k}) \epsilon_{t_i,k}^{(j)}$

end for

end for

3.3 提案法 2 : 局所線形 2 次近似を用いた近似法

本節では、局所線形ダイナミクスと 2 次コストに対する値関数が効率的に求められることを利用して、前回の最適軌道を中心に局所的な値関数の近似を行う解法を提案する。

3.3.1 局所線形 2 次近似

一般に非線形問題に対する値関数を求めるにはハミルトンヤコビ方程式を解かねばならず、その計算は困難な場合が多い。しかし、LQG 問題と呼ばれる、線形で連続な制御対象と、状態と入力 of 2 次形式で表されるコスト関数に対しては、値関数を状態の 2 次形式の形で効率的に求めることができることが知られている [9]。

そこで非線形な制御対象とコスト関数のある初期軌道まわりに局所線形近似し、この局所線形 2 次の問題に対して修正リカッチ方程式を解くことで、ある初期軌道周辺の値関数を求める方法が提案されている [2]。

制御対象である非線形確率システム

$$\dot{\mathbf{x}}_t = \mathbf{f}(\mathbf{x}_t, t) + \mathbf{g}(\mathbf{x}_t, t)\mathbf{u}_t + \mathbf{h}(\mathbf{x}_t, t)\epsilon_t$$

に対して、ある最適軌道周辺での局所線形ダイナミクスを考える。時刻 t_i における状態を $\mathbf{x}_{t_i}$ 、入力を $\mathbf{u}_{t_i}$ とすると、ノイズを考慮しないタイムステップ dt における状態の変化はオイラー法により次式のように表される。

$$\begin{aligned}\mathbf{x}_{t_{i+1}} &= \mathbf{x}_{t_i} + \left\{ \mathbf{f}(\mathbf{x}_{t_i}, t_i) + \mathbf{g}(\mathbf{x}_{t_i}, t_i)\mathbf{u}_{t_i} \right\} dt \\ &= \mathbf{x}_{t_i} + \mathbf{F}(\mathbf{x}_{t_i}, \mathbf{u}_{t_i}, t_i)dt\end{aligned}\tag{41}$$

いま、時刻 t_i における最適軌道上の状態を $\bar{\mathbf{x}}_{t_i}$ 、入力を $\bar{\mathbf{u}}_{t_i}$ とすると、サンプル軌道と最適軌道の入力の差分は、 $\delta\mathbf{u}_{t_i} := \mathbf{u}_{t_i} - \bar{\mathbf{u}}_{t_i}$ と定義できる。これらを (41) 式に代入すると、最適軌道周辺での局所線形ダイナミクスは次式のように書き直せる。

$$\bar{\mathbf{x}}_{t_{i+1}} + \delta\mathbf{x}_{t_{i+1}} = \bar{\mathbf{x}}_{t_i} + \delta\mathbf{x}_{t_i} + \mathbf{F}(\bar{\mathbf{x}}_{t_i} + \delta\mathbf{x}_{t_i}, \bar{\mathbf{u}}_{t_i} + \delta\mathbf{u}_{t_i}, t_i)dt\tag{42}$$

ここで $\mathbf{F}(\bar{\mathbf{x}}_t + \delta \mathbf{x}_t, \bar{\mathbf{u}}_t + \delta \mathbf{u}_t, t)$ に対して一次のテイラー近似を行うことで

$$\mathbf{F}(\bar{\mathbf{x}}_t + \delta \mathbf{x}_t, \bar{\mathbf{u}}_t + \delta \mathbf{u}_t, t) \simeq \mathbf{F}(\bar{\mathbf{x}}_t, \bar{\mathbf{u}}_t, t) + \frac{\partial \mathbf{F}}{\partial \mathbf{x}}|_{\bar{\mathbf{x}}_t} \delta \mathbf{x}_t + \frac{\partial \mathbf{F}}{\partial \mathbf{u}}|_{\bar{\mathbf{u}}_t} \delta \mathbf{u}_t \quad (43)$$

を得る．これを (42) に代入することで，次式で表される差分の従うダイナミクスが求められる．

$$\begin{aligned} \delta \mathbf{x}_{t_{i+1}} &= \left(\mathbf{I} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}}|_{\bar{\mathbf{x}}_t} dt \right) \delta \mathbf{x}_{t_i} + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{u}}|_{\bar{\mathbf{u}}_t} dt \right) \delta \mathbf{u}_{t_i} + (\mathbf{h}_{t_i} \sqrt{dt}) \xi_{t_i} \\ &= \mathbf{A}_{t_i} \delta \mathbf{x}_{t_i} + \mathbf{B}_{t_i} \delta \mathbf{u}_{t_i} + \mathbf{C}_{t_i} \xi_{t_i} \end{aligned} \quad (44)$$

ここで $\xi_t \sim N(0, \mathbf{I})$, $\delta \mathbf{x}_{t_0} = 0$ とする．

また，差分のダイナミクスと合わせて，軌道上の局所差分の 2 次コストも考える．最適軌道からの差分に関するコストをテイラー近似すると

$$\begin{aligned} r(\delta \mathbf{x}_{t_i} + \bar{\mathbf{x}}_{t_i}, \delta \mathbf{u}_{t_i} + \bar{\mathbf{u}}_{t_i}) dt \\ \simeq \left(r(\bar{\mathbf{x}}_{t_i}, \bar{\mathbf{u}}_{t_i}) + \frac{\partial r_{t_i}}{\partial \mathbf{x}} \delta \mathbf{x}_{t_i} + \frac{\partial r_{t_i}}{\partial \mathbf{u}} \delta \mathbf{u}_{t_i} \right. \\ \left. + \frac{1}{2} \delta \mathbf{x}_{t_i}^T \frac{\partial^2 r_{t_i}}{\partial \mathbf{x} \partial \mathbf{x}} \delta \mathbf{x}_{t_i} + \frac{1}{2} \delta \mathbf{u}_{t_i}^T \mathbf{R} \delta \mathbf{u}_{t_i} + \delta \mathbf{x}_{t_i}^T \frac{\partial^2 r_{t_i}}{\partial \mathbf{u} \partial \mathbf{x}} \delta \mathbf{u}_{t_i} \right) dt \end{aligned} \quad (45)$$

となる．

これによって (44), (45) は LQG 問題の形になるため，これを修正 LQG 問題として，修正リカッチ代数方程式を解くと，最適軌道周りの値関数は次式で与えられる．[10] (付録 A も参照のこと)

$$V(\delta \mathbf{x}_{t_i}, t_i) = s_t + \delta \mathbf{x}_{t_i}^T \mathbf{s}_{t_i} + \frac{1}{2} \delta \mathbf{x}_{t_i}^T \mathbf{S}_{t_i} \delta \mathbf{x}_{t_i} \quad (46)$$

$$s_{t_i} = q_{t_i} dt + s_{t_{i+1}} + \frac{1}{2} \sum_l \mathbf{h}_{l,t_i}^T \mathbf{S}_{t_{i+1}} \mathbf{h}_{l,t_i} dt - \frac{1}{2} \mathbf{d}_{t_i}^T \mathbf{W}_{t_i}^{-1} \mathbf{d}_{t_i} \quad (47)$$

$$\mathbf{s}_{t_i} = \nabla_{\mathbf{x}} q_{t_i} dt + \mathbf{A}_{t_i}^T \mathbf{s}_{t_{i+1}} - \mathbf{D}_{t_i}^T \mathbf{W}_{t_i}^{-1} \mathbf{d}_{t_i} \quad (48)$$

$$\mathbf{S}_{t_i} = \nabla_{\mathbf{xx}} q_{t_i} dt + \mathbf{A}_{t_i}^T \mathbf{S}_{t_{i+1}} \mathbf{A}_{t_i} - \mathbf{D}_{t_i}^T \mathbf{W}_{t_i}^{-1} \mathbf{D}_{t_i} \quad (49)$$

ここで,

$$\mathbf{d}_{t_i} := \mathbf{R}\mathbf{u}_{t_i}dt + \mathbf{B}_{t_i}^T \mathbf{s}_{t_{i+1}}, \quad (50)$$

$$\mathbf{W}_{t_i} := \mathbf{R}dt + \mathbf{B}_{t_i}^T \mathbf{S}_{t_{i+1}} \mathbf{B}_{t_i}, \quad (51)$$

$$\mathbf{D}_{t_i} := \mathbf{B}_{t_i}^T \mathbf{S}_{t_{i+1}} \mathbf{A}_{t_i}. \quad (52)$$

であり, $\mathbf{h}_{l,t_i}$ は $\mathbf{h}_{t_i}$ の l 列目の成分である.

3.3.2 提案法の着想

前節で述べた値関数は, 基準となる軌道周りでのダイナミクスに関するものなので, 前節の値関数の勾配に関して

$$\begin{aligned} \frac{\partial V(\delta \mathbf{x}_{t_i}, t_i)}{\partial \delta \mathbf{x}} \Big|_{\delta \mathbf{x}=0} &= \mathbf{s}_{t_i} \\ &\simeq -\lambda \nabla_{\mathbf{x}} \log \psi(\bar{\mathbf{x}}_{t_i}, t_i) \end{aligned} \quad (53)$$

と仮定すると, IPI で得られた j 回目の軌道周りの局所線形ダイナミクスを考えると, IPI の $j+1$ 回目の更新における時刻 t_i の値関数の微分は次式のように近似することができる.

$$\begin{aligned} \nabla_{\mathbf{x}} \log \psi^{(j)}(\mathbf{x}_{t_i}^{(j+1)}, t_i) &= \nabla_{\mathbf{x}} \log \psi^{(j)}(\bar{\mathbf{x}}_{t_i}^{(j)} + \delta \mathbf{x}_{t_i}, t_i) \\ &= \nabla_{\mathbf{x}} \log \psi^{(j)}(\bar{\mathbf{x}}_{t_i}^{(j)}, t_i) - \frac{1}{\lambda} \mathbf{S}_{t_i} \delta \mathbf{x}_{t_i} \end{aligned} \quad (54)$$

これを (31),(32) 式に代入することで, IPI におけるダイナミクスとコスト関数に対する補正項を近似的に計算することが出来る. 提案法の具体的な手順について Algorithm 2 に示す.

3.4 提案法 1 と 2 の比較

本章では, 前の章で述べた IPI の繰り返し計算上の問題について 2 種類の近似解法を提案した.

Algorithm 2 IPI-LQA

Input: $\{\epsilon_{t_1}, \epsilon_{t_2}, \dots, \epsilon_{t_N}\}$

Output: $\{\mathbf{u}_{t_1}, \mathbf{u}_{t_2}, \dots, \mathbf{u}_{t_N}\}$

Given: $\frac{\partial \log \psi^{(0)}(\mathbf{x}, t)}{\partial \mathbf{x}}, \mathbf{f}_t^{(0)}, q_t^{(0)}$

for $j = 1$ to J **do**

- $\mathbf{f}_t^{(j)} \Leftarrow \mathbf{f}_t^{(0)} + \lambda \nu_t \frac{\partial \log \psi^{(j-1)}(\mathbf{x}, t)}{\partial \mathbf{x}}$
- $q_t^{(j)} \Leftarrow q_t^{(0)} + \frac{\lambda^2}{2} \frac{\partial \log \psi^{(j-1)}(\mathbf{x}, t)}{\partial \mathbf{x}}^T \nu_t \frac{\partial \log \psi^{(j-1)}(\mathbf{x}, t)}{\partial \mathbf{x}}$

- create K roll-outs of the system from the same state $\mathbf{x}_0$

for $k = 1$ to K **do**

for $i = 1$ to N **do**

- $S(\tau_{i,k}) = \phi_{t_N, k} + \sum_{n=i}^N q_{t_n, k}^{(j)}$
- $P(\tau_{i,k}) = \frac{e^{-\frac{1}{\lambda} S(\tau_{i,k})}}{\sum_{k=1}^K e^{-\frac{1}{\lambda} S(\tau_{i,k})}}$

end for

end for

for $i = 1$ to N **do**

- $\mathbf{u}_{t_i} = \mathbf{R}^{-1} \mathbf{g}_t^T (\mathbf{h}_t \frac{\Sigma_\epsilon}{\lambda} \mathbf{h}_t^T)^{-1} \mathbf{h}_t \sum_{k=1}^K P(\tau_{i,k}) \epsilon_{t_i, k}^{(j)}$
- $\frac{\partial \log \psi_t^{(j)}}{\partial \mathbf{x}} = \lambda (\mathbf{h}_t \frac{\Sigma_\epsilon}{\lambda} \mathbf{h}_t^T)^{-1} \mathbf{h}_t \sum_{k=1}^K P(\tau_{i,k}) \epsilon_{t_i, k}^{(j)}$

end for

compute iLQG

for $i = 1$ to N **do**

$$\mathbf{S}_{t_i}^{(j)} = \nabla_{\mathbf{x}\mathbf{x}} q_{t_i}^{(j)} dt + \mathbf{A}_{t_i}^{T(j)} \mathbf{S}_{t_{i+1}}^{(j)} \mathbf{A}_{t_i}^{(j)} - \mathbf{D}_{t_i}^{T(j)} \mathbf{W}_{t_i}^{-1(j)} \mathbf{D}_{t_i}^{(j)}$$

end for

$$\frac{\partial \log \psi^{(j)}(\mathbf{x}, t)}{\partial \mathbf{x}} \Leftarrow \frac{\partial \log \psi_t^{(j)}}{\partial \mathbf{x}} + \frac{1}{\lambda} \mathbf{S}_t^{(j)} (\mathbf{x}_t - \bar{\mathbf{x}}_t)$$

end for

最近傍法を用いた近似解法である IPI-NN は、状態が近い場合の値関数が近いという仮定の下で複数の初期状態で PI を解き、その結果を用いて次の IPI 計算時に通る軌道上の値関数を、状態をクエリとした最近傍法によって近似する方法である。この方法は設定する初期状態の数が増えれば近似精度も向上することが期待できる。しかし、初期状態の数に対応した PI を解かなければならないので、もう一つの提案法である IPI-LQA に比べると必要なサンプル数は多くなる。そのため、多自由度問題への応用は IPI-LQA の方が有効であると考えられる。

一方、局所線形 2 次近似を用いた近似解法である IPI-LQA は、LQG 問題設定に対して値関数が効率的に求められることを利用して、前回の PI の軌道周りの局所的な値関数を求める方法である。この方法は、前述の IPI-NN に比べて計算効率が良いが、前回の最適軌道周辺での局所的な近似を行っているため、新しい生成軌道が前回の最適軌道から大きく外れる場合、近似誤差が大きくなるという問題が存在する。

4. シミュレーションによる評価

本章ではダブルスリット問題とロボットアームの2種類のシミュレーションを通して提案法の有効性を検証する.

4.1 ダブルスリット問題

ダブルスリットの通り抜けタスクのシミュレーションを通して, 提案法と既存手法である PI を比較し, 提案法の有効性を検証する.

4.1.1 問題設定

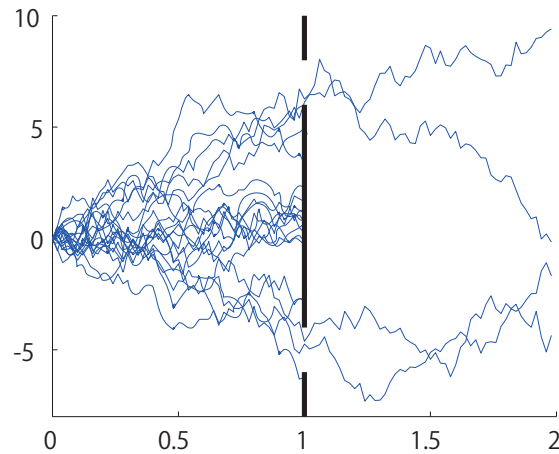


図 2 ダブルスリット問題

図 2 に示すような, ダブルスリットの通り抜け問題 [4] を用いて PI と 2 つの提案法の比較と評価を行う. この問題は最適入力を解析的に求めることが可能なため, 各手法の結果を最適解と比較して評価を行うことができる.

まず 2 つの状態変数 $x_t \in \mathbb{R}^1$, $y_t \in \mathbb{R}^1$ のダイナミクスを次式のように設定する.

$$\dot{x}_t = u_t \quad (55)$$

$$\dot{y}_t = 1 \quad (56)$$

これらの状態に対して逐次コスト $r(x_t, u_t, t)$ と、終端時間 $t_N = 2$ に対する終端コスト ϕ_{t_N} を次式で定める.

$$r(x_t, u_t, t) = q_t + 0.05u_t^2 \quad (57)$$

$$\phi_{t_N} = 0.5 \times x_{t_N}^2 \quad (58)$$

ただし $t = 1$ のとき,

$$q_t = \begin{cases} \infty & (x < -6, -4 < x < 6, 8 < x) \\ 0 & (\text{上記以外}) \end{cases} \quad (59)$$

であり、それ以外では $q_t = 0$ である.

このとき、ガウスノイズが無い場合の最適入力是一次式で与えられる.

$$u_t = \begin{cases} -4 & (0 \leq t < 1) \\ 4 & (1 \leq t < 2) \end{cases} \quad (60)$$

4.1.2 シミュレーション結果

各手法の結果の比較を表 1, 2, 3 及び、図 3, 4, 5, 6, 7, 8, 9, 10, 11 に示す. このシミュレーションでは IPI の感度パラメータを $\lambda = 15$ と設定した. また、各手法の合計サンプル数が等しくなるようにサンプリング回数を設定した.

表 1, 2, 3 はそれぞれノイズの共分散 Σ_ϵ を 1, 4, 9 ととった時の 200 回の試行における各手法の平均コスト、最小コスト、コストの分散を示している.

図 7, 9, 11 は各手法の時刻 t における最適入力、図 3, 4, 5 は各手法の実際の確率問題におけるダブルスリット上の軌道を表している. また、各手法の軌道の評価のため、図 6, 8, 10 に、ノイズ加わらない場合におけるダブルスリット上の軌道を示した. 図において、青線は IPI-LQA、黄線は IPI-NN、赤線は PI、緑線は解析解の結果をそれぞれ表している.

表 1, 2, 3 から、システムに付与されるノイズの大きさによって各手法の出力結果に大きな違いが現れていることが分かる. また、図 7, 9, 11 から、各手法

の入力と真の最適入力と比較でき、提案法が真の最適入力に近い値を得られていることが見て取れる．図 3, 4, 5 および 6, 8, 10 ではその入力に対する各手法の動作軌道が分かる．

表 1 各手法の結果比較, $\Sigma_{\epsilon} = 1$, (200 回試行)

コスト	PI	IPI-NN	IPI-LQA	最適解
平均	9.507	6.632	1.813	1.600
最小値	4.885	2.407	1.735	1.600
分散	5.016	44.628	29.600	-

$dt = 0.02$, IPI は $\lambda = 15$,

(サンプル数は, PI : 1000000.

IPI-NN : 10 (初期状態) $\times$ 10000 (PI のサンプル数)

$\times$ 10 (更新回数) = 1000000.

IPI-LQA : 100000 (PI のサンプル数)

$\times$ 10 (更新回数) = 100000.)

表 2 各手法の結果比較, $\Sigma_{\epsilon} = 4$ (他の設定に関しては Table1 と同様)

コスト	PI	IPI-NN	IPI-LQA	最適解
平均	1.835	2.462	2.120	1.600
最小値	1.752	2.115	2.060	1.600
分散	0.001	0.018	0.0007	-

表 3 各手法の結果比較, $\Sigma_{\epsilon} = 9$ (他の設定に関しては Table1 と同様)

コスト	PI	IPI-NN	IPI-LQA	最適解
平均	∞	2.619	2.300	1.600
最小値	∞	1.966	2.244	1.600
分散	-	0.079	0.0005	-

4.1.3 考察

図 3, 4, 5 の結果から, 提案法, 従来法共に下側のスリットを通る軌道を導出できていることが分かる. 一方, 確率ノイズの影響を排除して, シミュレーション結果を見ると, 図 6, 8 から, ガウスノイズが小さい場合では, どの手法もダブルスリット下側を通る最適軌道を導出できていることが分かる. しかし, 表 3 を見ると, PI は最適制御入力を得られていないのに対し, IPI-LQA, IPI-NN は制御入力を生成できていることが分かる. また, 表 1, 2, 3 の結果から, IPI-LQA が IPI=NN に比べ, 平均コストが低く, 分散の小さい結果を得ていることが分かる. これらの結果から, 同一サンプル数での計算において IPI-LQA がより分散の小さい, 安定した制御入力を得られていることがいえる.

本シミュレーションによって提案法が, PI がうまく動作しないような状況でも適切な制御入力を導出できることが示せた. また, IPI-LQA に比べ, IPI-NN がコストの分散が大きいため, IPI-LQA の方がより安定して制御入力を導出できる方法であると考えられる.

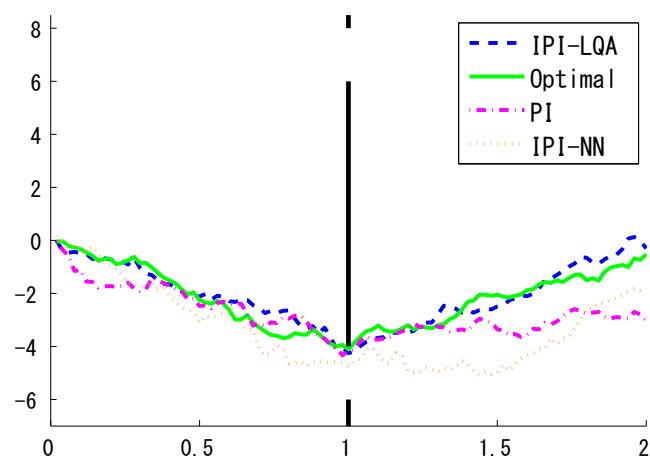


図 3 ダブルスリット ($\Sigma_\epsilon = 1$)

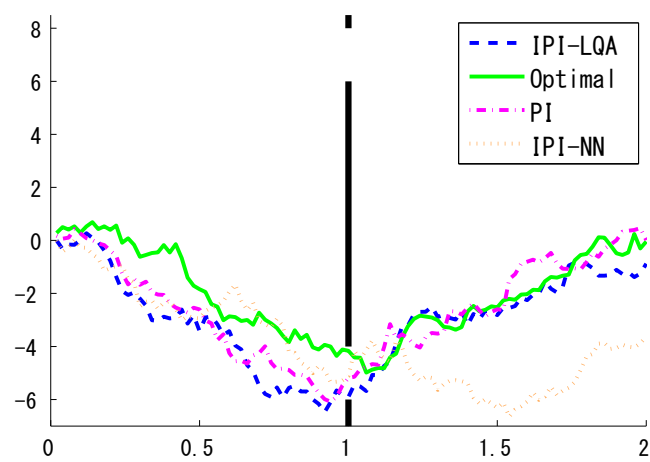


図 4 ダブルスリット ($\Sigma_\epsilon = 4$)

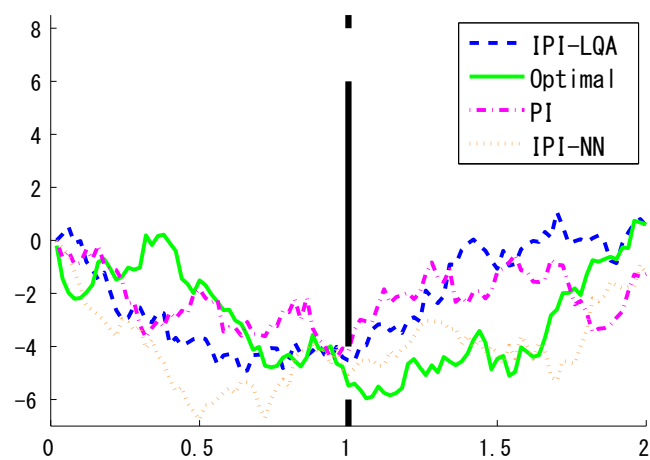


図 5 ダブルスリット ($\Sigma_{\epsilon} = 9$)

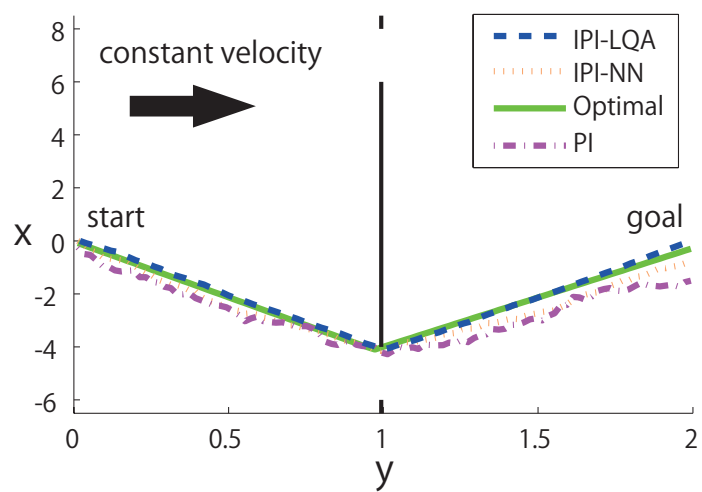


図 6 ダブルスリット ($\Sigma_{\epsilon} = 1$): 確率ノイズなし

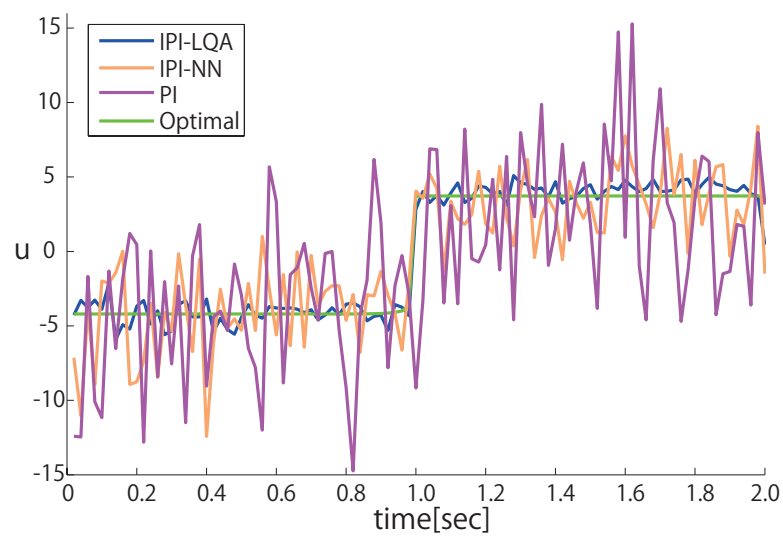


図 7 入力 ($\Sigma_{\epsilon} = 1$)

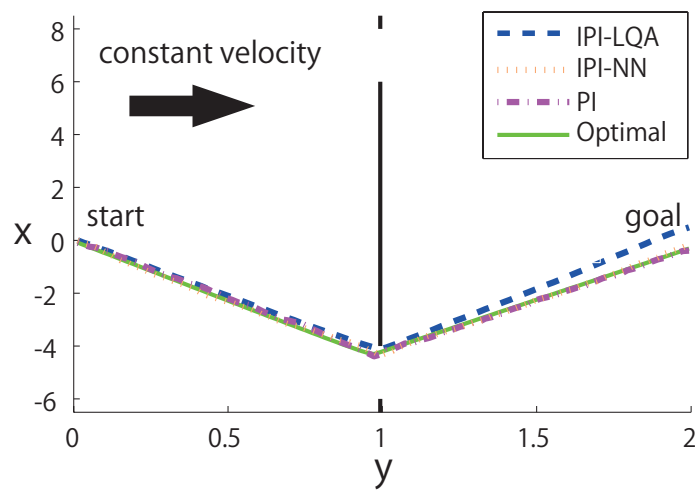


図 8 ダブルスリット ($\Sigma_{\epsilon} = 4$): 確率ノイズなし

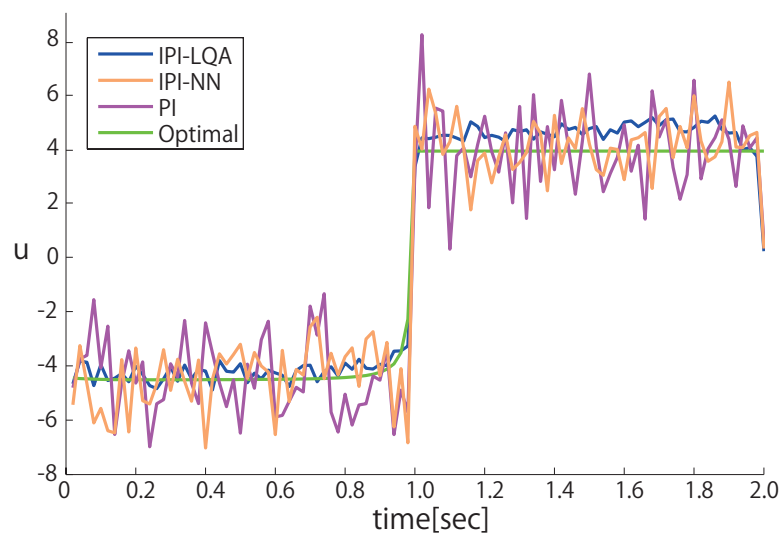


図 9 入力 ($\Sigma_{\epsilon} = 4$)

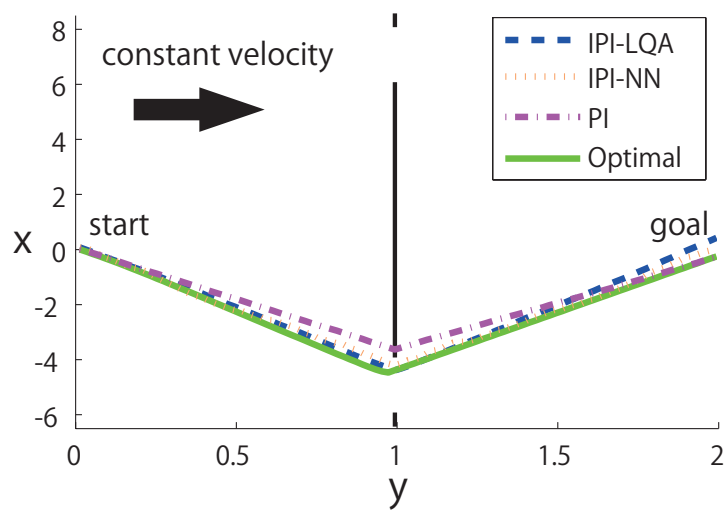


図 10 ダブルスリット ($\Sigma_{\epsilon} = 9$) : 確率ノイズなし

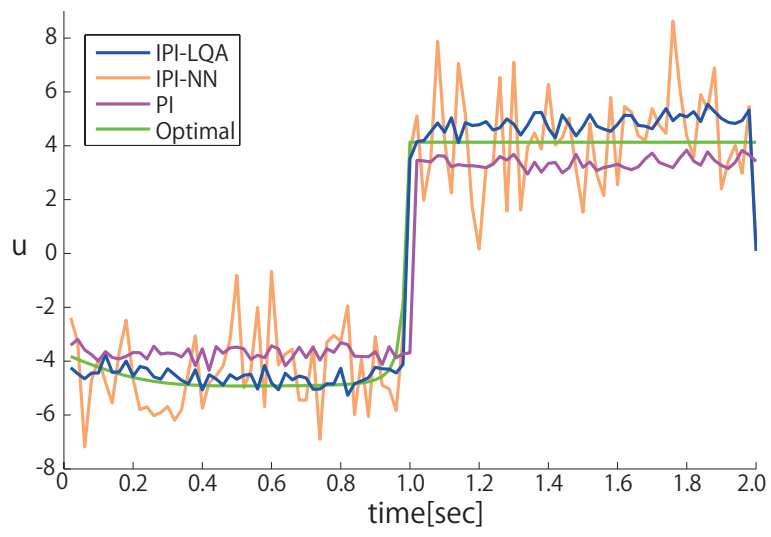


図 11 入力 ($\Sigma_{\epsilon} = 9$)

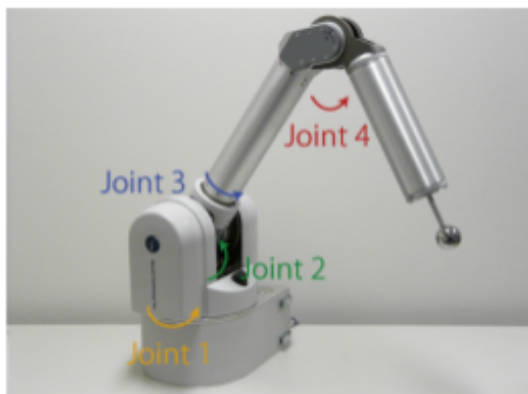


図 12 WAM Arm 実機

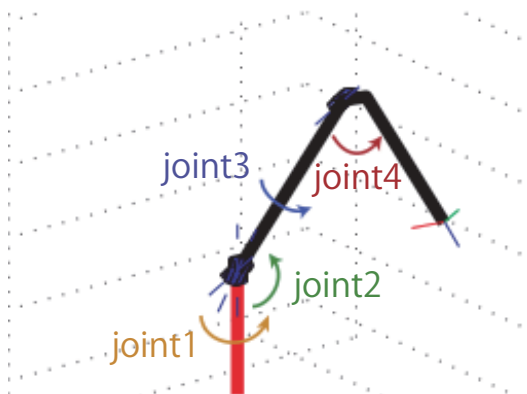


図 13 WAM Arm シミュレーション

4.2 ロボットシミュレータへの適用

本論文で提案した近似解法のうち，IPI-NN は複数の初期状態を必要とし，実問題，特に高次元問題に対しては適用が困難だと考えられる．しかし，もう一方の提案手法である IPI-LQA は実問題への適用が IPI-NN に比べ容易であると考えられる．

そこで本節では 4 関節 WAM Arm のシミュレータを用いて提案法である IPI-LQA を用いたロボット制御について検討する．

4.2.1 4 関節バレットシミュレータ

ダブルスリットによる比較の他，IPI-LQA の多次元での応用に関して確認するためにシミュレータへの実装を行った．

シミュレーションの対象として，冗長マニピュレータ 4 関節ロボットアームである WAM Arm(Barrett Technology 社製)(図 12) を考え，Robotics Toolbox[11] を用いて作成されたシミュレータを使用した (図 13)．なお，物理パラメータには Barrett 社によって公開されている仕様を反映させた．

また、本シミュレーションではシステムの入力に共分散行列 $\mathbf{I}$ のガウスノイズを付与することで、システムに不確定性のある確率システムとした。

今回は joint4 の関節角及び、関節角速度の制御タスクを行った。

実行するタスクはアームの振り上げタスクとし、逐次コストと終端コスト (終端時刻 $t_N = 2.5[\text{sec}]$) を次式のように設定した。

$$r_{t_i} = \frac{1}{2} \mathbf{x}_{t_i}^T \mathbf{x}_{t_i} + \frac{1}{2} \mathbf{R} \mathbf{u}_{t_i}^2 \quad (61)$$

$$\phi_{t_N} = \frac{1}{2} \mathbf{x}_{t_N}^T \mathbf{x}_{t_N} \quad (62)$$

ここで $\mathbf{x}_{t_i} = \begin{pmatrix} \theta_{t_i} \dot{\theta}_{t_i} \end{pmatrix}^T$ であり、 θ_{t_i} は時刻 t_i での関節角、 $\dot{\theta}_{t_i}$ は角速度である。また、ノイズの共分散行列 $\Sigma_\epsilon = \mathbf{I}$ 、入力重み $\mathbf{R} = 0.5$ 、パラメータ $\lambda = 1$ とし、初期状態を $\theta_{t_0} = -2.618[\text{rad}]$ 、 $\dot{\theta}_{t_0} = 0$ とした。

これによって角速度と目標角度との差を最小化するようなタスクを実行する。

また、PI を多自由度アームなどの剛体ダイナミクスに対して適用する場合、システムの摩擦等の要因により、システムの動作しやすい方向に偏ったサンプルが多く得られてしまい、状態空間全体に一樣なサンプルが得られない問題がある。そのため、本研究では Theodorou ら [12] が提案した逆動力学制御器を用いた PI の実装を基に、IPI のシミュレーションを行った。

4.2.2 シミュレーション結果

シミュレーションの結果を図 14, 16, 18 に示す。結果は 5000 サンプル、6 回更新後の結果である。図 14 は導出された最適入力を加えた時の実空間でのアームの動作軌道を表している。図 16 は各時刻における関節角度の変化を表し、図 18 は角速度の変化を表している。図 16, 18 から、角度、角速度が終端時刻で 0 に近くなり、目的を達成できていることが分かる。

また比較のため、IPI-LQA の補正を加えない場合の結果を図 15, 17, 19 に示す。前述の図と同様、図 15 はアームの動作軌道、図 17 は関節角度の変化、図 19 は角速度の変化を表している。図 20 は両手法の反復回数とコストの関係である。

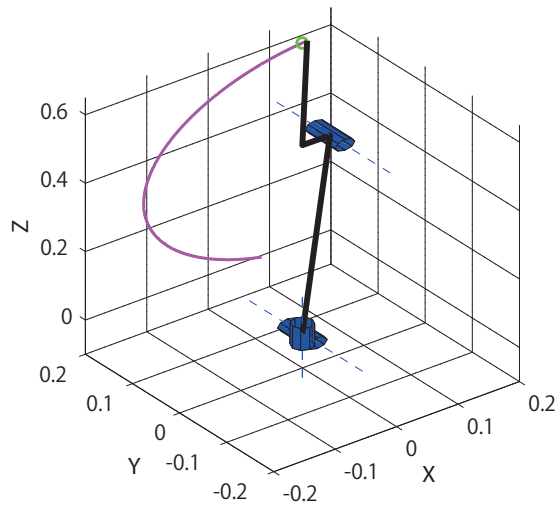


図 14 アーム軌道 (IPI-LQA)

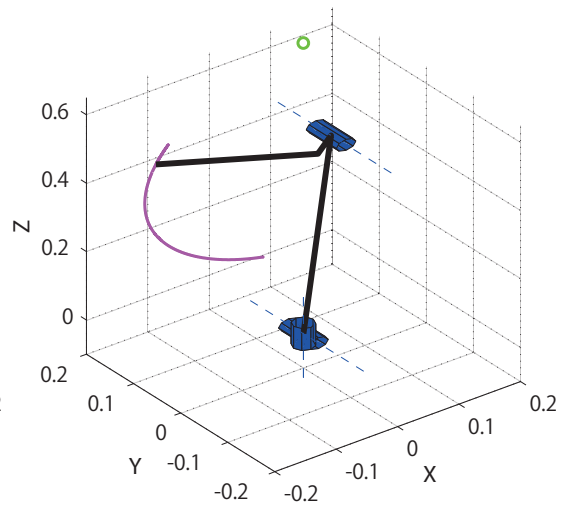


図 15 アーム軌道 (LQA 補正なし)

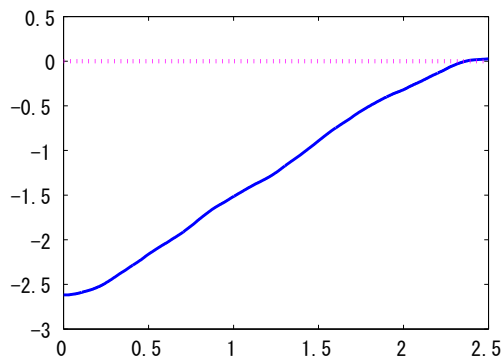


図 16 関節角度 (IPI-LQA)

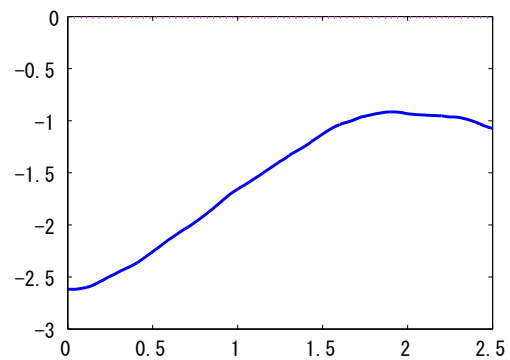


図 17 関節角度 (LQA 補正なし)

4.2.3 考察

本章では提案法である IPI-LQA の 2 次元システムへの応用としてロボットアームの振り上げタスクを行った．本章の結果から，提案法が 2 次元の問題に対しても適用可能であることが示された．また，LQA による補正を加えない場合との比較から，本手法が近似器として正しく機能していることが示せた．

しかし，IPI はサンプリングの際のダイナミクスとコストに対して補正項を加

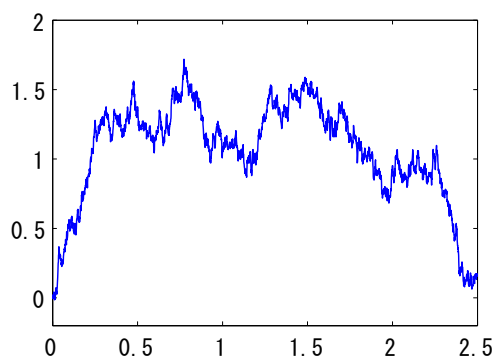


図 18 関節角速度 (IPI-LQA)

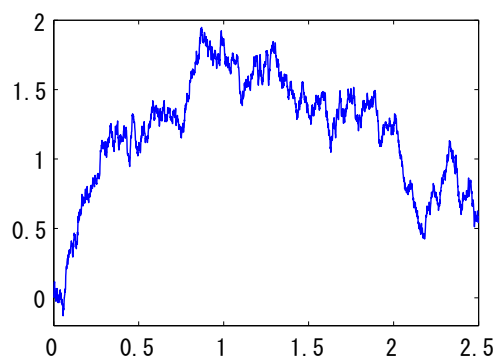


図 19 関節角速度 (LQA 補正なし)

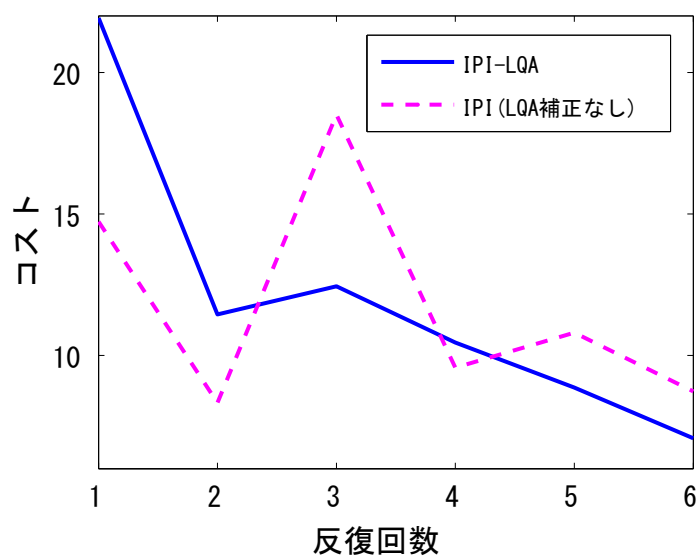


図 20 学習曲線

えるため、次元が増加すると IPI-LQA における仮定である局所的な近似の範囲を大きく越える場合が多くなるため、近似誤差が増大して適用が困難になると考えられる。そのため、より高次元な問題に対する適用には、値関数の微分に関する精度の向上が課題であると考えられる。

5. おわりに

本研究のまとめと今後の課題について述べる.

本研究では, 近年提案された非線形確率最適制御法の有効な解法である反復型軌道積分最適制御法に対して, 数値計算上の問題点を指摘し, 最近傍探索と局所線形近似を用いた2つの近似解法を提案した. 提案手法では, IPI の繰り返し計算における入れ子構造を解消するため, 関数近似を用いて更新式を求めることで, 入れ子構造を回避した効率的な近似解法の構築を目指した. また, シミュレーションによる検証を行い, 提案手法の有効性を確認した.

本提案手法により, 実世界でのロボット制御など, 不確定性の強い状況において対象を制御するための技術の発展に貢献できると考えられる. また, 今後不確定性の強い状況における制御法が発展することで, 環境の変化に対しても強い自動制御の発展とロボットの活動範囲の拡大が期待できる.

今後の課題について述べる. 本研究では, 非線形確率最適制御問題の関数近似を用いた解法を扱ったが, ロボットなどの高次元, 複雑な問題への対応を考えると, さらなる近似精度の向上とサンプリングの効率化が必要であると考えられる.

サンプリングの効率化に関しては, モンテカルロサンプリングにおける効率化の手法である重点サンプリング [13] や, その他のサンプリング手法が利用できると考えられる. また, Atkeson らは文献 [14] で動的計画法における値関数の近似のための効率的なサンプリング手法に関して論じており, この方法は各状態点毎にサンプリングを考えるため, 提案法の一つである IPI-NN と相性が良いと考えられる.

また, PI を基とした強化学習法として PI^2 と呼ばれる手法 [5][15][16] が提案されており, 学習効率や実装の単純さからロボット制御などの分野で注目されている. そのため, PI のクラス拡大である IPI と PI^2 を組み合わせた強化学習法の構築にも, 提案法は貢献できると考える.

謝辞

はじめに本研究を進めるにあたり，知能システム制御研究室の杉本謙二教授には研究のご指導に留まらず，研究者，技術者としてあるべき姿勢，物事に対する考え方など，様々な面でご指導，ご鞭撻を頂きました．心よりお礼を申し上げます．

また，お忙しい中，副指導教員を引き受けてくださり，ゼミナール等で大変貴重なご意見，ご提案を頂きました数理情報学研究室の池田和司教授に深く感謝致します．

松原崇充助教には研究，私生活共にご多忙な時期にも関わらず，本研究を進めうえでの議論に加えて，研究の進め方や論文の書き方に至るまで，貴重なご助言，細部に渡る熱心なご指導を賜りました．改めて厚く御礼申し上げます．

平田健太郎准教授，並びに，小木曽公尚助教には，定例研究会などにおいて鋭いご指摘や的確なアドバイスを頂き，至らない私をサポートして頂きました．ここに深く感謝致します．

福岡大学の野田賢教授（元知能システム制御研究室 准教授）には異なる観点からの貴重なご意見に加え，学生生活全般に関してもアドバイスを頂いたことに対して，心より感謝致します．

日々の研究生活において多くの事務的な手続きを引き受けてくださいました秘書の林英子さんに感謝いたします．

また，研究に留まらず，学生生活においても多大なサポートをして頂いた知能システム制御研究室の諸氏に厚く御礼申し上げます．

最後になりましたが，研究，学生生活の両面で様々なアドバイスを頂いた先輩方，研究生活を陰で支えて頂いた両親，ならびに友人に心より感謝致します．

参考文献

- [1] セバスチャン・スラン, ウルフラム・バーガード, ディーター・フォックス, 上田隆一. 確率ロボティクス. 毎日コミュニケーションズ, 2007.
- [2] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback. In *proceedings of the American Control Conference*, pp. 300–306, 2005.
- [3] 大住晃. 確率システム入門. 朝倉書店, 2002.
- [4] H. J. Kappen. Path integrals and symmetry breaking for optimal control theory. *J. Statical Mechanics : Theory and Experiment (11)*, 2005. P11011.
- [5] E. A. Theodorou, J. Buchli, and S. Schaal. A generalized path integral control approach to reinforcement learning. *Journal of Machine Learning Research*, Vol. 11, pp. 3137–3181, 2010.
- [6] 佐藤訓志, H. J. Kappen, 佐伯正美. 軌道積分に基づく非線形確率最適制御の解法. 第12回計測自動制御学会制御部門大会予稿集, 2012. P0194.
- [7] B. K. Oksendal. *Stochastic Differential Equations, An Introduction with Applications*. Springer-Berlin, New York, 6th edition, 2003.
- [8] C. M. ビショップ, 元田浩, 栗田多喜夫, 樋口知之, 松本裕治, 村田昇. パターン認識と機械学習 上 - ベイズ理論による統計的予測. 丸善出版, 2007.
- [9] E. Todorov and W. Li. Iterative linear quadratic regulator design for non-linear biological movement systems. In *proceedings of the American Control Conference*, pp. 222–229, 2004.
- [10] 長井英生. 確率微分方程式. 共立出版株式会社, 1999.
- [11] P. I. Corke. Robotics toolbox for matlab. *IEEE Robotics & Automation Magazine*, Vol. 3(1), pp. 24–32, 1996.

- [12] E. A. Theodorou, J. Buchl, and S. Schaal. Path integral-based stochastic optimal control for rigid body dynamics. In *proceedings of Adaptive Dynamic Programming and Reinforcement Learning*, pp. 219–225, 2009.
- [13] H. Hachiya, T. Akiyama, M. Sugiyama, and J. Peters. Adaptive importance sampling for value function approximation in off-policy reinforcement learning. *Neural Networks*, Vol. 22, No. 10, pp. 1399–1410, 2009.
- [14] C. G. Atkenson and B. J. Stephens. Random sampling of states in dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics part B*, Vol. 38, No. 4, 2008.
- [15] E. A. Theodorou, J. Buchli, and S. Schaal. Tendon-driven control of biomechanical and robotic systems: A path integral reinforcement learning approach. In *proceedings of International Conference on Robotics and Automation*, pp. 208–214, 2012.
- [16] E. A. Theodorou, E. Todorov, and F. J. Valero-Cuevas. Neuromuscular stochastic optimal control of a tendon driven index finger model. In *proceedings of the American Control Conference*, pp. 348–355, 2011.

付録

A. LQG 制御問題の詳細計算

ここでは 3.3.1 の LQG 問題に対する解法の詳細について説明する.

以下のような局所線形ダイナミクスとコスト関数で定義される LQG 問題を考える.

$$\delta \mathbf{x}_{t_i+1} = \mathbf{A}_{t_i} \delta \mathbf{x}_{t_i} + \mathbf{B}_{t_i} \delta \mathbf{u}_{t_i} + \mathbf{C}_{t_i} \xi_{t_i} \quad (63)$$

$$\begin{aligned} \text{cost}_{t_i} = & \left(r(\bar{\mathbf{x}}_{t_i}, \bar{\mathbf{u}}_{t_i}) + \frac{\partial r_{t_i}}{\partial \mathbf{x}} \delta \mathbf{x}_{t_i} + \frac{\partial r_{t_i}}{\partial \mathbf{u}} \delta \mathbf{u}_{t_i} \right. \\ & \left. + \frac{1}{2} \delta \mathbf{x}_{t_i}^T \frac{\partial^2 r_{t_i}}{\partial \mathbf{x} \partial \mathbf{x}} \delta \mathbf{x}_{t_i} + \frac{1}{2} \delta \mathbf{u}_{t_i}^T \mathbf{R} \delta \mathbf{u}_{t_i} + \delta \mathbf{x}_{t_i}^T \frac{\partial^2 r_{t_i}}{\partial \mathbf{u} \partial \mathbf{x}} \delta \mathbf{u}_{t_i} \right) dt \end{aligned} \quad (64)$$

いま, 時刻 t_i における状態誤差 $\delta \mathbf{x}_{t_i}$ に対する値関数を $v(\delta \mathbf{x}_{t_i})$ とすると, 逐次コストと終端コストの形から

$$v(\delta \mathbf{x}_{t_{N-1}}) = s_{t_{N-1}} + \delta \mathbf{x}_{t_{N-1}}^T \mathbf{s}_{t_{N-1}} + \frac{1}{2} \delta \mathbf{x}_{t_{N-1}}^T \mathbf{S}_{t_{N-1}} \delta \mathbf{x}_{t_{N-1}} \quad (65)$$

が成り立つ. このとき, 1 ステップ前の値関数 $v(\delta \mathbf{x}_{t_{N-2}})$ は

$$\begin{aligned} v(\delta \mathbf{x}_{t_{N-2}}) = & \min_{\delta \mathbf{u}_{t_{N-2}}} \left\{ \left(r(\bar{\mathbf{x}}_{t_{N-2}}, \bar{\mathbf{u}}_{t_{N-2}}) + \frac{\partial r_{t_{N-2}}}{\partial \mathbf{x}} \delta \mathbf{x}_{t_{N-2}} + \frac{\partial r_{t_{N-2}}}{\partial \mathbf{u}} \delta \mathbf{u}_{t_{N-2}} \right. \right. \\ & + \frac{1}{2} \delta \mathbf{x}_{t_{N-2}}^T \frac{\partial^2 r_{t_{N-2}}}{\partial \mathbf{x} \partial \mathbf{x}} \delta \mathbf{x}_{t_{N-2}} + \frac{1}{2} \delta \mathbf{u}_{t_{N-2}}^T \mathbf{R} \delta \mathbf{u}_{t_{N-2}} \\ & \left. \left. + \delta \mathbf{x}_{t_{N-2}}^T \frac{\partial^2 r_{t_{N-2}}}{\partial \mathbf{u} \partial \mathbf{x}} \delta \mathbf{u}_{t_{N-2}} \right) dt \right\} + \mathbb{E} \left[v(\delta \mathbf{x}_{t_{N-1}}) \right] \end{aligned} \quad (66)$$

となる. 最適制御入力を π とすると, 値関数は

$$\begin{aligned} v(\delta \mathbf{x}_{t_{N-2}}) = & \left(r(\bar{\mathbf{x}}_{t_{N-2}}, \bar{\mathbf{u}}_{t_{N-2}}) + \frac{\partial r_{t_{N-2}}}{\partial \mathbf{x}} \delta \mathbf{x}_{t_{N-2}} + \frac{\partial r_{t_{N-2}}}{\partial \mathbf{u}} \pi \right. \\ & + \frac{1}{2} \delta \mathbf{x}_{t_{N-2}}^T \frac{\partial^2 r_{t_{N-2}}}{\partial \mathbf{x} \partial \mathbf{x}} \delta \mathbf{x}_{t_{N-2}} + \frac{1}{2} \delta \pi^T \mathbf{R} \pi + \delta \mathbf{x}_{t_{N-2}}^T \frac{\partial^2 r_{t_{N-2}}}{\partial \mathbf{u} \partial \mathbf{x}} \pi \Big) dt \\ & + \mathbb{E} \left[v(\delta \mathbf{x}_{t_{N-1}}) \right] \end{aligned} \quad (67)$$

である．上記の期待値の項は

$$\begin{aligned}
\mathbb{E}[v(\delta \mathbf{x}_{t_{N-1}})] &= \mathbb{E}[v(\mathbf{A}_{t_{N-2}} \delta \mathbf{x}_{t_{N-2}} + \mathbf{B}_{t_{N-2}} \pi + \mathbf{C}_{t_{N-2}} \xi_{t_{N-2}})] \\
&= s_{t_{N-1}} + (\mathbf{A}_{t_{N-2}} \delta \mathbf{x}_{t_{N-2}} + \mathbf{B}_{t_{N-2}} \pi)^T \mathbf{s}_{t_{N-1}} \\
&\quad + \frac{1}{2} (\mathbf{A}_{t_{N-2}} \delta \mathbf{x}_{t_{N-2}} + \mathbf{B}_{t_{N-2}} \pi)^T \mathbf{S}_{t_{N-1}} (\mathbf{A}_{t_{N-2}} \delta \mathbf{x}_{t_{N-2}} + \mathbf{B}_{t_{N-2}} \pi) \\
&\quad + \frac{1}{2} \sum_l \mathbf{h}_{l,t_{N-2}}^T \mathbf{S}_{t_{N-1}} \mathbf{h}_{l,t_{N-2}}
\end{aligned} \tag{68}$$

となる．したがって (67) 式に (68) 式を代入すると

$$\begin{aligned}
v(\delta \mathbf{x}_{t_{N-2}}) &= r(\bar{\mathbf{x}}_{t_{N-2}}, \bar{\mathbf{u}}_{t_{N-2}}) dt + s_{t_{N-1}} + \frac{1}{2} \sum_l \mathbf{h}_{l,t_{N-2}}^T \mathbf{S}_{t_{N-1}} \mathbf{h}_{l,t_{N-2}} \\
&\quad + \delta \mathbf{x}_{t_{N-2}}^T \left(\frac{\partial r_{t_{N-2}}}{\partial \mathbf{x}} dt + \mathbf{A}_{t_{N-2}} \mathbf{s}_{t_{N-1}} \right) \\
&\quad + \delta \mathbf{x}_{t_{N-2}}^T \left(\frac{\partial^2 r_{t_{N-2}}}{\partial \mathbf{x} \partial \mathbf{x}} dt + \mathbf{A}_{t_{N-2}}^T \mathbf{S}_{t_{N-1}} \mathbf{A}_{t_{N-2}} \right) \delta \mathbf{x}_{t_{N-2}} \\
&\quad + \pi^T (\mathbf{d}_{t_{N-2}} + \mathbf{D}_{t_{N-2}} \delta \mathbf{x}_{t_{N-2}}) + \frac{1}{2} \pi^T \mathbf{W}_{t_{N-2}} \pi
\end{aligned} \tag{69}$$

t_{N-1} から t_0 まで再帰的にこれが成り立つので，結局この問題における値関数は

$$v(\delta \mathbf{x}_{t_i}) = s_{t_i} + \delta \mathbf{x}_{t_i}^T \mathbf{s}_{t_i} + \frac{1}{2} \delta \mathbf{x}_{t_i}^T \mathbf{S}_{t_i} \delta \mathbf{x}_{t_i} \tag{70}$$

と表せる．いま

$$\mathbf{d}_{t_i} := \mathbf{R} \mathbf{u}_{t_i} dt + \mathbf{B}_{t_i}^T \mathbf{s}_{t_{i+1}}, \tag{71}$$

$$\mathbf{W}_{t_i} := \mathbf{R} dt + \mathbf{B}_{t_i}^T \mathbf{S}_{t_{i+1}} \mathbf{B}_{t_i}, \tag{72}$$

$$\mathbf{D}_{t_i} := \mathbf{B}_{t_i}^T \mathbf{S}_{t_{i+1}} \mathbf{A}_{t_i}. \tag{73}$$

とおいている．

最適制御入力 π は勾配を 0 にする入力なので $\delta u_{t_i} = -\mathbf{W}_{t_i}^{-1} \mathbf{d}_{t_i} - \mathbf{W}_{t_i}^{-1} \mathbf{D}_{t_i} \delta \mathbf{x}_{t_i}$ である．したがって

$$s_{t_i} = q_{t_i} dt + s_{t_{i+1}} + \frac{1}{2} \sum_l \mathbf{h}_{l,t_i}^T \mathbf{S}_{t_{i+1}} \mathbf{h}_{l,t_i} dt - \frac{1}{2} \mathbf{d}_{t_i}^T \mathbf{W}_{t_i}^{-1} \mathbf{d}_{t_i} \tag{74}$$

$$\mathbf{s}_{t_i} = \nabla_{\mathbf{x}} q_{t_i} dt + \mathbf{A}_{t_i}^T \mathbf{s}_{t_{i+1}} - \mathbf{D}_{t_i}^T \mathbf{W}_{t_i}^{-1} \mathbf{d}_{t_i} \quad (75)$$

$$\mathbf{S}_{t_i} = \nabla_{\mathbf{xx}} q_{t_i} dt + \mathbf{A}_{t_i}^T \mathbf{S}_{t_{i+1}} \mathbf{A}_{t_i} - \mathbf{D}_{t_i}^T \mathbf{W}_{t_i}^{-1} \mathbf{D}_{t_i} \quad (76)$$

となる.