未知システムに対する適応出力最適制御

Adaptive Output Optimal Control Algorithm for Unknown System Dynamics Based on Policy Iteration

東京工業大学 O.著者 大竹 進, 共著 山北 昌毅
S.Ohtake and M.Yamakita
Dep. of Mechanical and Control Eng., Tokyo Institute of Technology

Abstract In this paper, first we propose an algorithm of adaptive control scheme for systems whose dynamics are unknown and it is shown that the control converges to the approximate optimal state feedback control solution. In the algorithm, we introduce integrators in front of unknown systems, in order to transform the systems into an augmented ones whose input dynamics is known, then, we use policy iteration to find an approximate optimal control input. Second, we introduce two extended algorithms, one is an adaptive output optimal control algorithm for systems which are unknown and only output is measurable, and another one is an adaptive $H_{\infty}$ sub-optimal control for linear systems.

提案手法の有効性は、数値例により示す。

2 HJB 方程式

以下の非線形時不変アフィニングシステムを対象とする。

$$
\dot{x}(t) = f(x) + g(x)u, \quad x(0) = x_0 \quad (1)
$$

$$
y(t) = h(x) \quad (2)
$$

ここで、$x(t) \subseteq \mathbb{R}^n$, $u(t) \subseteq \mathbb{R}$ であり、上式はリプシッツ連続であると仮定する。さらに、状態フィードバック則を $u = \mu(x)$ とし、無限区間の評価関数を以下のように設定する。

$$
V(x(t)) = \int_t^\infty r(x(\tau), u(\tau))d\tau \quad (3)
$$

$$
r(x, u) := x^T Q x + \mu(x)^T R \mu(x) \quad (4)
$$

ただし、$Q, R$ は正定値とする。

最適制御問題は、(2) 式を最小化するようなフィードバック解を見つけることに帰着する。そこで、以下のハミルトニアンを導入する。

$$
H(x, u, V_x) = r(x, u) + (V_x)^T (f(x) + g(x)u) \quad (5)
$$

ここで、$V_x$ は、関数 $V$ の $x$ での偏微分を意味する。HJB 方程式より、関数 $V^*(x), u^*(x)$ がそれぞれ、最適評価関数、最適入力となる必要十分条件は、次の式を満たすことである。

$$
0 = \min_{u^*} [H(x, u^*, V_x^*)] \quad (6)
$$

$$
u^*(x) = -\frac{1}{2} R^{-1} g^T(x) V_x^*(x) \quad (7)
$$
3 Policy Iteration

(6),(7) 式の HJB 方程式を解くことによって、最適入力が求まるが、一般的にこれは難しく、オフラインで解くには、$f(x), g(x)$ が必要となる。しかし、オンラインで解を導出する方法として、[1] により、以下の policy iteration が提案された。

1. 次式の $V^{(i)}(x)$ を解く (policy evaluation)
   $$V^{(i)}(x(t)) = \int_t^{t+T} r(x(\tau), \mu^{(i)}(x(\tau)))d\tau + V^{(i)}(x(t+T))$$
   (8)

2. policy の更新 (policy improvement)
   $$\mu^{(i+1)}(x) = -\frac{1}{2} R^{-1} g^T(x) V^{(i)}(x)$$
   (9)

ただし、$\mu^{(i)}, V^{(i)}$ は、i 回アップデートされた後の状態フィードバック値と、コスト関数を意味し、$\mu^{(i)}(x)$ は安定化コントローラであると仮定する。上記の policy iteration において、以下の定理が成立する。

Theorem.1 [1],[2] Policy Iteration によって得られる状態フィードバック値は、最適入力に収束する。

ただし、(9) 式において、入力の特性関数 $g(x)$ を用いて居ることに注意する必要がある。

4 Extended Policy Iteration

3 章にて紹介した policy iteration では、入力の特性関数 $g(x)$ を用いて居ることに問題があった。システムのダイナミクスを用いずに適応最適制御を行うために、入力 $u$ を加えた新しい状態変数 $x' := [x^T, u^T]^T$ と、新たな入力変数 $v = \dot{u}$ を定義し、(1) 式を以下のようにシステムを拡大する。

$$\dot{x}_a := \begin{bmatrix} \dot{x} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} f(x) + g(x)u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} v$$

$$\dot{x}_a := f_a(x_a) + g_a(x_a)v$$
   (10)

ここで、状態フィードバック値を、$v = \mu_a(x_a)$ とおく。次に、(10) 式に対応する評価関数を以下のように定義する。ただし、$Q_a = \text{diag}(Q, R), R_a$ は正定とする。

$$V_a(x_a(t)) = \int_t^{t+T} r_a(x_a(\tau), v(\tau))d\tau$$
   (11)

$$r_a(x_a,v) := x_a^T Q_a x_a + v^T R_a v$$
   (12)

ただし、$V_a$ のうち、$x_a$ の成分を $J_x, v$ の成分を $J_v$ と定義する。この拡大系に対して、以下の拡大 policy iteration を行う。

1. 次式の $V_0^{(i)}(x_a)$ を解く (policy evaluation)
   $$V_0^{(i)}(x_a(t)) = \int_t^{t+T} r_a(x_a(\tau), \mu_0^{(i)}(x_a(\tau)))d\tau + V_0^{(i)}(x_a(t+T))$$
   (13)

2. policy の更新 (policy improvement)
   $$\mu_0^{(i+1)}(x_a) = -\frac{1}{2} R_0^{-1} g_0^T(x_a) V_0^{(i)}(x_a)$$
   (14)

3. 初期入力の更新
   $$u^{(i+1)} = \left\{ u \mid \frac{\partial V_0^{(i)}(x_a)}{\partial u} \bigg|_{x_a=[x^T, u^T]} = 0 \right\}$$
   (15)

ただし、$\mu_0^0(x_a)$ は安定化コントローラであると仮定する。ここで、(10) 式より、$g_0(x_a)$ は既定となることに注意すると、

Proposition.1 拡大 Policy Iteration により、$\mu_0^{(\infty)}$ は、拡大系システム (10) 式の最適フィードバック解に収束する。

Proposition.2 (12) 式の評価関数 $R_a \to 0$ とすると、$V_0(x_a) \to V(x)$ となり、拡大 Policy Iteration により得られた入力 $u$ は、システムの最適入力に漸近する。

これより、未知システム (1) の最適入力を求めることができる。

5 ニューラルネット

式 (13) 的 Policy Evaluation を解くことは難しいので、ニューラルネットにより、近似解を導出する。コスト関数を以下のように近似する。

$$V_0^{(i)}(x_a) = \sum_{j=1}^L w_j^{(i)} \phi_j(x_a) = (W_L^{(i)})^T \Phi_L(x_a)$$
   (16)

ここで、$L$ はニューロンの数、$\phi_j(x_a) \in C^1, \phi_j(0) = 0$ は、アクティベーション関数、$w_j^{(i)}$ は、各アクティベーション関数の重みであり、$W_L^{(i)}, \Phi_L(x_a)$ は、それぞれをまとめたベクトルである。これを、式 (13) に代入すると、以下のようになる。

$$W_L^{(i)}^T \Phi_L(x_a(t)) = \int_t^{t+T} r_a(x_a, \mu_0^{(i)}(x_a))d\tau + W_L^{(i)}^T \Phi_L(x_a(t+T))$$

第52回自動制御連合講演会
2009年11月21, 22日(大阪大学)
上式のように Neural net で近似したとき、一般的に、以下のような残差が残る。

\[
\delta_L^{(i)}(x(t)) = \int_t^{t+T} r_a(x_a, \mu_a^{(i)}(x_a)) \, dt + W_L^{(i)}(\Phi_L(x_a(t + T)) - \Phi_L(x_a(t))
\]

この \( \delta_L(x_a) \) を 0 にする \( W_L^{(i)} \) は、以下のように計算できる。

\[
u_L^{(i)} = - \Lambda^{-1} \Xi \tag{17}
\]

ただし、\( \Lambda, \Xi \) は、それぞれ、

\[
\Xi = \begin{bmatrix}
\int_t^{t+T} r_a(x_a, \mu_a^{(i)}(x_a)) \, dt \\
\int_{t+T}^{t+2T} r_a(x_a, \mu_a^{(i)}(x_a)) \, dt \\
\vdots \\
\int_{t+L-T}^{t+LT} r_a(x_a, \mu_a^{(i)}(x_a)) \, dt
\end{bmatrix}
\]

\[
\Lambda = \begin{bmatrix}
\Phi_L(x_a(t + T)) - \Phi_L(x_a(t)) \\
\Phi_L(x_a(t + 2T)) - \Phi_L(x_a(t + T)) \\
\vdots \\
\Phi_L(x_a(t + (L - 1)T)) - \Phi_L(x_a(t + (L - 1)T))
\end{bmatrix}
\]

ただし、ある正の定数 \( \epsilon \) をもうけ、常に \( \delta_L(t) < \epsilon \) が成立立つ場合、この計算を行わないとする。これは、\( \delta_L(t) \) が小さく、現在の状態フィードバック則が最も近い場合には、まえに \( \nu_L^{(i)} \) が発散する誤勤を起こすためである。

4.5 章をまとめてみると、最終的に採用される Policy Iteration は、以下のようになる。

1. 次式的 \( V_a^{(i)}(\hat{x}_a) \) を解く (policy evaluation)

\[
W_L^{(i)} \Phi_L(x_a(t)) = \int_t^{t+T} r_a(x_a(\tau), \mu_a^{(i)}(x_a(\tau))) \, d\tau + W_L^{(i)} \Phi_L(x_a(t + T)) \tag{18}
\]

2. policy の更新 (policy improvement)

\[
\mu_a^{(i+1)}(x_a) = - \frac{1}{2} R_a^{-1} g_a^T(x_a) \left( \frac{\partial \Phi_L(x_a)}{\partial x} \right)^T W_L^{(i)} \tag{19}
\]

3. 初期入力の更新

\[
u^{(i+1)} = \left\{ u \mid \left. W_L^{(i)} \frac{\partial \Phi_L(x_a)}{\partial u} \right|_{x_a=[x, u]} = \right\} \tag{20}
\]

以降、この Extended Policy Iteration を基準とした、2つの発展型アルゴリズムを紹介する。

6 発展型 1: 適応最適出カフィードバック

6.1 オブザーバ設計

5 章までの提案手法では、システムが未知であるにもかかわらず、状態が全て観測できるという、実用上難しい仮定に基づくものであった。そのため、以下のような、システムの特性を用いないオブザーバを併用することを考える。

Lemma.1 [3] システムがドリフト可観測であるならば、状態 \( x \) を、\( z = [y, \dot{y}, \cdots, y^{(n)}]^T \) に座標変換したシステムに対して、

\[
\dot{z} = A z + K (y - \dot{y}) + k_d P C^T \text{sgn}(y - \dot{y}) \tag{21}
\]

\[
y = \dot{z}_1 \tag{22}
\]

とすると、\( z - \dot{z} \) は渐近安定である。ただし、

\[
A = \begin{bmatrix}
0 & 1 & \cdots & 0 \\
0 & 0 & \ddots & 0 \\
\vdots & 0 & \ddots & 0 \\
0 & 0 & \cdots & 0
\end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 \end{bmatrix}
\]

\[
F = \begin{bmatrix} 0 \end{bmatrix}
\]

\[
(A - KC) P + P (A - KC)^T + BB^T + FF^T + 2 \alpha P + \gamma^2 T < 0 \tag{23}
\]

を満たし、\( k_d, \alpha, \gamma \) は、正の定数である。

上記のオブザーバを用いると、状態 \( x \) を推定することはできないが、\( x \) を座標変換した \( z \) に関しては、推定することができる。よって、この章ではこれ以降、状態 \( z \) に座標変換したシステムに関して考えていく。

6.2 数値シミュレーション

6.2.1 座標変換後のシステム

対象の未知システムとして以下のものを考える。

\[
\dot{x} = \begin{bmatrix} -x_1^3 - x_2 \\ x_1 + x_2 
\end{bmatrix} + \begin{bmatrix} 1 \\
0
\end{bmatrix} u \tag{24}
\]

\[
y = \begin{bmatrix} 1 \\ 0 \end{bmatrix} x \tag{25}
\]

ただし、初期状態 \( x_0 = [0.1, 0.1]^T \) とする。これを、状態 \( z = [y, \dot{y}]^T \) に状態変換すると、

\[
\dot{z} = \begin{bmatrix} z_2 \\ -3z_1^2 z_2 - z_1 + z_2
\end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u \tag{26}
\]

\[
y = \begin{bmatrix} 1 \\ 0 \end{bmatrix} z \tag{27}
\]

このシステムに対して、評価関数を以下のように設定する。ただし、\( Q, R \) は単位行列とする。

\[
V_z = \int_0^\infty (z^T Q z + u^T R u) \, dt
\]
ニューロンの数を 15 とし、アクティベーション関数と初期重みを、以下のように設定した。

\[ \Phi_L(z) = [z_1^2, z_1z_2, z_2^2, z_1^3, z_1^2z_2, z_1z_2^2, z_2^3, z_1z_2^2, z_1^2z_2^2, z_1^2z_2, z_1z_2, z_2]^T \]

\[ W_L^{(0)} = [2, 0, 2, 0_{1 	imes 12}]^T \]

さらに、Policy Evaluation のパラメータ \( T = 0.3 \) とした。この条件で、入力の特性を既知として Policy Iteration を行った様子が、図 4.2 である。図 4.2 は、Policy Evaluation によって得られた重み \( W_L^{(i)} \) の変化を示すグラフで、図 2 は、初期コントローラと、最終コントローラの応答の比較をしたものである。このときのコスト評価値は、\( V_z = 5.2217 \times 10^{-2} \) となった。

### 6.2.2 拡大システム

(26) 式に対して、拡大系を (10) のように定義し、\( V_a, r_a \) を、(11), (12) 式のように定義する。今回は、ニューロンの数を 21 とし、アクティベーション関数と、初期重みは、以下のよう設定した。

\[ \Phi_L(z_a) = [z_1^2, z_1z_2, z_1z_3, z_1z_2z_3, z_2^2, z_1z_2^2, z_2z_3, z_1z_2z_3z_4, z_1z_2^2z_3, z_2^3, z_1z_2z_3z_4z_5, z_1z_2^2z_3z_4, z_2^4, z_1z_2z_3z_4z_5z_6, z_2^3z_3, z_2z_3^2, z_2z_3^2z_4, z_2^2z_3z_4, z_2z_3^2z_4z_5, z_2^3z_3z_4z_5z_6]^T \]

\[ W_L^{(0)} = [0, 0, 0, 0, -20, -5, 0_{1 	imes 15}]R_a \]

さらに、評価値 \( R_a = 0.1 \) 、シミュレーションパラメータ \( \epsilon = 1.0 \times 10^{-4} \) とし、システムのダイナミクスは未知だが、状態が直線を続けたとする拡大 Policy Iteration を行った。このときの初期コントローラと、最終コントローラの応答は、図 3 のようになり、評価値は、\( V_z = 5.2257 \times 10^{-2} \) となり、ほぼ前節の結果に漸近した。

### 6.2.3 オプサーバーとの併用系

状態を観測できないとし、\( z \) を \( \hat{z} \) に置き換えて Extended Policy Iteration を行う。オプサーバーのパラメータとして、

\[ \alpha = 0.01, \ \gamma = 4.0, \ \kappa_d = 0.01 \]

\[ \text{eig}(A - KC) = [-220, -210] \]

さらに、\( R_a = 0.1, \ \epsilon = 0.002 \) とした。その他のパラメータは、前節と同じとした。このときの初期、最終コントローラの応答は、図 4 のようになった。評価値は、\( V_z = 5.2783 \times 10^{-2} \) となり、最適値と比べると、誤差の目立つ結果となったが、初期コントローラ (\( V_z = 5.6474^{-2} \)) よりは改善が見られる。

### 7 発展型 2：適応 \( H_{\infty} \) 準最適制御

#### 7.1 線形システムに対する適応制御

対象を以下の線形時不変システムに限定する。

\[ x(t) = Ax + Bu(x(t)), \ \ x(0) = x_0 \]  \( (28) \)

線形システムのコスト関数は 2 次式で表されることを用いると、(8),(9) 式の policy iteration は以下のよう表せる。

1. 次式の \( P^{(i)} \) を解く (policy evaluation)

\[ x^T(t)P^{(i)}x(t) = \int_0^{t+T} r(x(\tau), \mu^{(i)}(x(\tau))) \ d\tau \]

\[ + x^T(t + T)P^{(i)}x(t + T) \]  \( (29) \)

2. policy の更新 (policy improvement)

\[ \mu^{(i+1)}(x) = -R^{-1}B^T P^{(i)}x \]  \( (30) \)

また、(13)〜(15) の Extended Policy Iteration は、

\[ \begin{bmatrix} \dot{x} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} x_a + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v \]  \( (31) \)

\[ := A_a x_a + B_a v \]  \( (32) \)

と拡大系を定義すると、

1. 次式の \( P_a^{(i)} \) を解く (policy evaluation)

\[ x_a^T(t)P_a^{(i)}x_a(t) = \int_0^{t+T} r_a(x_a(\tau), \mu_a^{(i)}(x_a(\tau))) \ d\tau \]

\[ + x_a^T(t + T)P_a^{(i)}x_a(t + T) \]  \( (33) \)

2. policy の更新 (policy improvement)

\[ \mu_a^{(i+1)}(x_a) = -R_a^{-1}B_a^T P_a^{(i)}x_a \]  \( (34) \)

3. 初期入力の更新

\[ u_0 = (u \mid \frac{\partial x_a^T P_a x_a}{\partial u} |_{x_a^T = |x_{a1}^T, u_1^T|} = 0) \]  \( (35) \)

**Proposition 3** (33)〜(35) 式において、\( R_a \rightarrow 0 \) とするとき、\( \mu_a^{(n)} \) による入力 \( u \) は、拡大前のシステムの最適入力に漸近する。さらに、

\[ \mu_a^{(n)} = -R_a^{-1}B_a^T P_a^{(n)} x_a \]  \( (36) \)

\[ := \begin{bmatrix} Fx & Fu \end{bmatrix} x \]  \( (37) \)
と定義すると、

$$-R^{-1}B^TP^{(\infty)} = -F_{u^{-1}}Fx$$  (38)

$$P^{(\infty)} = \begin{bmatrix} I & -(F_{u^{-1}}Fx)^T \end{bmatrix} P^{(\infty)}_{a} \begin{bmatrix} I & -F_{u^{-1}}Fx \end{bmatrix}$$  (39)

が成立する．つまり，拡大前のシステムの最適フィードバック解、最適コスト関数を求めることができる．

### 7.2 適応 $H_{\infty}$ 準最適アルゴリズム

対象となるシステムを以下とする。

$$\dot{x} = Ax + Bu + Kd$$  (40)

$$z = \begin{bmatrix} Cx \end{bmatrix}$$  (41)

$d$ は外乱で、$A, B$ は未知、$C, K$ は既知とする．すると $H_{\infty}$ 準最適制御とは、$d$ から $z$ までの $H_{\infty}$ ノルムを $\gamma$ 以下に抑え、かつ下式を満たすシステムを安定化するフィードバック解 $u = K_zdx$ を求めることがある.

$$V_h = \min u \sup_d \int_0^\infty (\|y\|^2 + \|u\|^2 - \gamma^2 \|d\|^2) d\tau$$  (42)

このような $u$ が存在するかどうかの必要十分条件は、以下を満たす正定な $P_h$ が存在することであり、(42) 式を満たすフィードバック入力は、

$$A^TP_h + P_hA + P_h(\frac{1}{\gamma}KK^T - BB^T)P_h + C^TC = 0$$  (43)

$$u = -B^TP_hx$$

で与えられる．これに対して、前章で用いていた最適制御において、ARE 方程式は、

$$A^T P + PA - PBR^{-1}B^TP + Q = 0$$  (44)

$$u = -R^{-1}B^TPx$$  (45)

であった．つまり、

$$R = I, \quad Q = C^TC + \frac{1}{\gamma^2}PKK^T P$$  (46)

において、$P, P_h$ の両者は等しくなる．さらに、(43)(44)に関して、次の結果がいえる．

Proposition.4 $P^{(0)} = 0$ とし、$P^{(i)}$ を、以下のように従って更新する。

$$A^TP^{(i+1)} + P^{(i+1)}A - P^{(i+1)}BB^TP^{(i+1)} + \frac{1}{\gamma^2}P^{(i)}KK^TP^{(i)} + C^TC = 0$$  (47)

すると、$P^{(\infty)} = P_h$ となる。

この Proposition と、7.1 章で述べた結果を元に、以下の $H_{\infty}$ 準最適適応アルゴリズムを提案する。

---

### step.0 $P^{(0)} = 0$ とおく．安定化フィードバック則 $u = K_zdx$ が既知

### step.1 $P^{(i)}$ に基づき、$Q = C^TC + \frac{1}{\gamma^2}P^{(i)}KK^TP^{(i)}$, $R = 1$ と設定する。

### step.2 7.1 章 Extended policy iteration と、(39) 式より、$P^{(i+1)}$ を計算し、step.1 に戻る。

このアルゴリズムにより得られた入力はすなわち $H_{\infty}$ 準最適フィードバック入力となる．さらに、このときの $P_h, K_h$ も導出することができる。

**Remark** 上記システムでは、外乱の影響を受けてない状態値が観測できると仮定していることに注意。

### 7.3 数値シミュレーション

対象システムを以下とする。

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1 & 0 \end{bmatrix} d$$  (48)

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} x$$  (49)

評価関数は、$Q, R$ を (46) 式のように設定し、$R_v = 0.008$ と定めた。このシステムに対して提案した適応 $H_{\infty}$ 準最適適応アルゴリズムを適用すると、$P^{(i)}$ は図 5 のように更新され最終的に、

$$P^{(\infty)} = \begin{bmatrix} 1.3096 & 0.4006 \\ 0.4006 & 0.1421 \end{bmatrix}, \quad P_h = \begin{bmatrix} 1.3084 & 0.4002 \\ 0.4002 & 0.1419 \end{bmatrix}$$

となり、十分な精度が得られることが分かる。

### 8 Conclusion

本論文では、未知で、非線形なシステムに対して、初期安定化コントローラを、最適コントローラに接近させる適応制御法を提案し、シミュレーションによって、その有効性を検証した．さらに、発展型として、出力のみが観測可能なシステムに対する適応最適出力フィードバック制御、状態観測可能な線形システムに対する適応準最適 $H_{\infty}$ 制御の 2 つを提案した．今後の課題としては、外乱の考慮を強化する。

| Table 1: Cost Value [10^{-2}] |
|-----------------------|------------|---|---|
| System | $J_z$ | $J_v$ | error |
| 6.2.1 | 5.2217 | 0 | 0 |
| 6.2.2 | 5.2257 | 0.2259 | 0.0040 |
| 6.2.3 | 5.2783 | 0.3503 | 0.0526 |
図 1: Policy Update

図 2: Response of system 8.1

図 3: Response of system 8.2

図 4: Response of system 8.3

図 5: Adaptation of $P^{(i)}$

参考文献


