

# Just-In-Time法によるむだ時間倒立振り子の安定化制御

## Stabilization of Time-Delayed Inverted Pendulum Control Systems using Just-In-Time Method

福田 憲一郎, 牛田 俊, 出口 光一郎

Ken-ichiro Fukuda, Shun Ushida, Koichiro Deguchi

東北大学 大学院情報科学研究科

Graduate School of Information Sciences, Tohoku University

キーワード: 倒立振り子(Inverted pendulum), 人間の運動制御(Human motor control),  
むだ時間(Time delay), Just-In-Time手法(Just-In-Time method)

連絡先: 〒980-8579 仙台市青葉区荒巻字青葉6-6-01  
東北大学 大学院情報科学研究科 システム情報科学専攻 出口・岡谷研究室  
福田憲一郎, Tel./Fax: (022)795-7017, E-mail: fukuda@fractal.is.tohoku.ac.jp

### 1 はじめに

人間が行う正確で滑らかな運動制御は、視覚、聴覚、体性感覚など様々なセンサ情報を適切に組み合わせることによって実現されている。一方、現在開発されている二足歩行ロボットや産業用ロボットの場合も数多くのセンサ情報に基づいて動作している。しかし、学習や予測のような高度な知能をもつ人間と、プログラム通りに動作するロボットを比較すれば、前者が圧倒的に高い制御性能を発揮し、様々な環境の変化にも対処する能力を有しているのは明らかである。このように両者は、構造や制御メカニズム、その振舞い等に相違点が多い。しかしながら、運動制御の根底にあるフィードバック制御のレベルに注目すると、信号伝達のむだ時間が存在するという両者の共通点に気づく。特に、人間の運動制御系には神経系の信号伝達や、脳内の情報処理に起因する数十ミリ～数百ミリ秒の大きな時間遅れが存在する。ロボットの制御系では、カメラからの画像情報に基づくビジュアルサーボ系において、むだ時間の影響が顕著に現れる。すなわち、カメラのフレームレートの低さや画像の膨大な情報処理に起因する時間遅れが、制御系の性能を著しく低下させる。本研究の目的の一つは、むだ時間制御系をカメラ-ロボットアームに応用した実機実験を通じて、むだ時間の存在が制御系設計に与える影響を定量的に評価することである。また、人間はどのようにしてむだ時間を克服しているであろうかという疑問に答えることで、人間の運動制御メカニズムを工学的見地から

解釈し再現することも本研究の目的の一つとする。

本論文では、人が行うある特定のタスクに焦点をあて、問題を簡単化する。そのタスクとは、倒立振り子の安定化制御である。人間が視覚により手の上で棒を立てることを考えると、視神経の信号伝達や脳内の情報処理に大きな時間遅れが存在するにもかかわらず、短時間の練習によりある程度の時間の直立状態を維持できる。一方、倒立振り子制御装置による振り子の安定化は、次節で述べるようにロータリエンコーダの角度情報に基づく簡単なフィードバック制御で実現可能である。しかし、人間のむだ時間を含む運動制御系を模擬するため、ロータリエンコーダの代わりにカメラからの視覚情報を振り子の角度センサとして用いると、カメラの処理系に起因する大きなむだ時間によって安定化制御が困難になる。本システムを用いることで、むだ時間の克服という人の運動制御における本質的な問題を扱うことが可能となる。

人間はどのようにしてむだ時間を克服しているであろうかという疑問に答えるために、Cabreraらの研究グループは人の制御メカニズムとその振舞いを考え、人の拳動を統計学的な側面から詳しく調べている<sup>1, 2, 3)</sup>。彼らは、数多くの被験者に対して人が倒立振り子の安定化を試みた場合の拳動のデータを収集し、確率システムによるモデル化を試みている。また、Cabreraらのモデルの欠点を修正し、むだ時間と状態依存ノイズの関係について制御理論的な側面からの議論および数値シミュレーションを行った研究も報告されている<sup>4, 5)</sup>。これらの研究が何らかの数学モデルを利用している

のに対して、本論文では人の運動制御メカニズムへのアプローチとして、Just-In-Time(以下、JIT)手法を用いて人間の学習・記憶による意思決定機構を再現する。JIT手法では、複雑な非線形システムの入出力データベースが対象の動特性を表現する手段であり、特定の数学モデルを用いない点が上記の研究とは大きく異なる。JIT手法がもつ振舞いの記憶という特性に着目し、むだ時間を含む系の同定と振子の安定化制御を同時に行い、倒立振子安定化制御系の構築を目指す。

## 2 カメラ-ロボットアームによる倒立振子の安定化制御

### 2.1 視覚を用いた倒立振子制御系

Fig. 1に示すような水平面を動く2リンク非線形Direct Driveアーム(以下、DDアーム)と、その先端に付いた自由関節をもつ倒立振子、そして角度検出を行うためのカメラからなるシステムを考える。(X Y Z), (u v)はそれぞれ、ワールド座標系、画像座標系を表す。

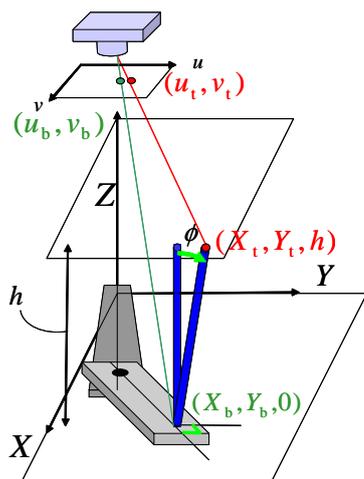


Fig. 1 カメラ、DDアーム、倒立振子の関係

視覚により振子の角度 $\phi$ を計測することを考える。振子の長さ $h$ は既知であるとし、振子の先端が画像上の $(u_t, v_t)$ に写ったとする。振子の角度 $\phi$ は微小であると仮定すると、振子先端は三次元空間内の高さ $h$ の平面上を移動する。ワールド座標上の点 $(X_t, Y_t, h)$ と画像上の点 $(u_t, v_t)$ との関係は、透視投影行列 $P$ を用いて次のように表される<sup>6)</sup>。

$$w \begin{pmatrix} u_t \\ v_t \\ 1 \end{pmatrix} = P \begin{pmatrix} X_t \\ Y_t \\ h \\ 1 \end{pmatrix} \quad (1)$$

したがって、 $P$ が既知であるカメラを用いて振子先端を撮影し、画像座標 $(u_t, v_t)$ を得ることにより、

振子先端の三次元位置 $(X_t, Y_t, h)$ を求めることができる。同様に振子根元のワールド座標 $(X_b, Y_b, 0)$ も、振子根元をカメラで撮影することにより、画像から求めることができる。 $(X_t, Y_t, h)$ と $(X_b, Y_b, 0)$ より、振子の角度 $\phi$ は次式で示される。

$$\phi \approx \sin \phi = \frac{\sqrt{(X_b - X_t)^2 + (Y_b - Y_t)^2}}{h} \quad (2)$$

### 2.2 これまでの実験的事実

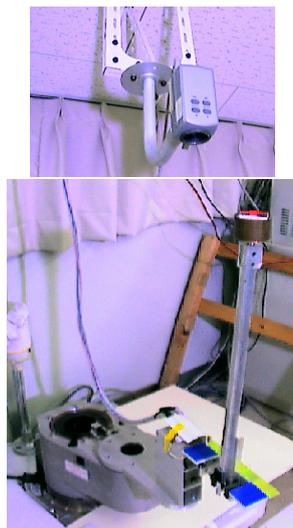


Fig. 2 実際のカメラ、DDアームおよび倒立振子

本実験では問題を簡単にするために、振子の運動をDDアーム2軸目先端の回転軌道の接線方向に拘束する。また、DDアームは1軸目は固定し、2軸目のみを使用する。実験に使用したDDアーム及び倒立振子をFig. 2に示す。振子の角度検出機構として、高速かつ高精度に動作するロータリエンコーダ系と、人の視覚に相当するカメラ-画像処理系を、Fig. 3に示すように切換えて用いる。Fig. 4は、振子の角度検出をロータリエンコーダ系とカメラ-画像処理系によって同時に行った結果である。ロータリエンコーダによる角度情報は1[ms]毎に更新されるのに対し、視覚による角度情報は、カメラのフレームレート30[Hz]が原因で、約33[ms]毎の信号である。さらに、画像の取得に1~2フレーム分(33~66[ms])のむだ時間が存在することもわかる。

このシステムにおいて、振子の角度の比例ゲイン、微分ゲイン、DDアーム2軸の角度の比例ゲイン、微分ゲインをそれぞれ $(P_p, D_p, P_2, D_2)$ として、これらを調整することにより倒立振子の安定化実験を行った結果、以下のような実験的事実が得られた。

- (1) ロータリエンコーダで振子の角度検出を行い、適切なゲインを設定することにより容易に安定化が実現された。

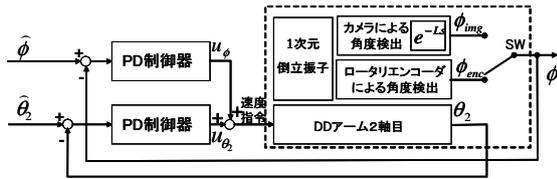


Fig. 3 スwitching機構をもつ倒立振り子安定化制御系

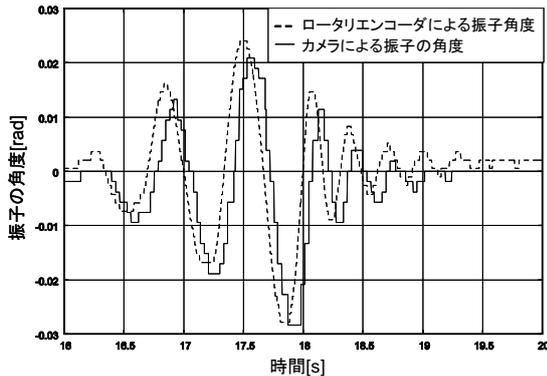


Fig. 4 ロータリエンコーダと視覚情報の比較

- (2) ロータリエンコーダから得られた振り子の角度情報の時間分解能をカメラと同じ30[Hz]に0次ホールドした場合でも、(1)と同様のゲインで容易に安定化は実現された。
- (3) ロータリエンコーダから得られた振り子の角度情報に50[ms]のむだ時間を挿入した場合、適切なゲインを設定することで、振り子が収束も発散もしない特定の周期運動をすることが確認された。漸近安定化は達成されていない。

### 2.3 数値シミュレーションによるゲインの調整

前節で、カメラ-画像処理に起因するむだ時間が倒立振り子の安定化制御を難しくしていることを述べた。そこで、Fig. 5に示す視覚によるDDアーム-倒立振り子系のシミュレータを作成した。シミュレーションによって、むだ時間を含むDDアーム-倒立振り子系を安定化するPDゲインの存在範囲を確認することができる。

倒立振り子と速度制御モードのDDアームの運動方程式は倒立振り子、DDアームの角度をそれぞれ  $\phi$ 、 $\theta_2$  として次式で示される。

$$(I_p + m_p r_p^2) \ddot{\phi} + m_p l_2 r_p \cos \phi \ddot{\theta}_2 + d_p \dot{\phi} - m_p g r_p \sin \phi = 0 \quad (3)$$

$$\alpha \ddot{\theta}_2 + \beta \dot{\theta}_2 = \gamma u \quad (4)$$

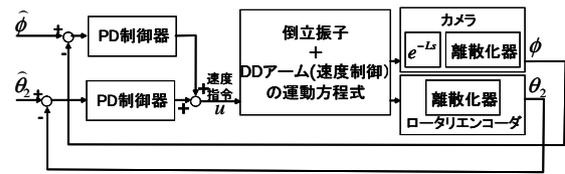


Fig. 5 視覚を用いた倒立振り子系のシミュレータ

DDアーム2軸は振り子に対して質量が大きく、大トルクを発生させることができるので、振り子の挙動はDDアームのダイナミクスには影響しないと仮定した。各記号に対する意味をTable 1に示す。式(3)、(4)を用いて、DDアームに速度指令値  $u$  を入力すると倒立振り子の角度  $\phi$  とDDアームの角度  $\theta_2$  を出力するようなシミュレータを構成した。倒立振り子の角度  $\phi$  はカメラで検出することを前提としているため、むだ時間  $\tau = 0.05$ [s] を挿入し、30[Hz] で離散化した。また、DDアームの角度  $\theta_2$  はDDアーム本体に取り付けられたロータリエンコーダで検出することができるため、1[kHz] で離散化を行った。またその他のパラメータの値についてはTable 1に示したとおりである。Fig. 5のシミュレータを用いて、振り子とDDアームのゲイン ( $P_p, D_p, P_2, D_2$ ) を様々に変化させることにより、安定化できるゲインを探索した。ここで言う安定化とは、十分時間が経った後、アームの角度も振り子の角度もある範囲内に収まっていることである。

結果をFig. 6に示す。\*印がシミュレータ上で安定化できたゲインである。Fig. 6の結果では、DDアームの比例ゲイン  $P_2 = 0.1$  に固定とした。むだ時間を含むFig. 5の系にも安定化を可能にするPDゲインが存在することを確認し、また膨大な4次元パラメータ空間の中で安定化ゲインのある程度の位置を知ることができた。

Table 1 シミュレーションに用いたパラメータ一覧

振り子	
$m_p = 0.871$ [kg]	質量
$l_p = 0.522$ [m]	長さ
$r_p = 0.459$ [m]	根元から重心までの距離
$I_p = 0.0113$ [kg · m <sup>2</sup> ]	重心周りの慣性モーメント
$d_p = 0.02$ [ $\frac{N \cdot m \cdot s}{rad}$ ]	振り子の粘性摩擦係数
DDアーム2軸	
$l_2 = 0.35$ [m]	長さ
$\alpha = 1.75 \times 10^{-2}$ [-]	速度制御のパラメータ
$\beta = 4.49 \times 10^{-1}$ [-]	速度制御のパラメータ
$\gamma = -8.93 \times 10^{-1}$ [-]	速度制御のパラメータ

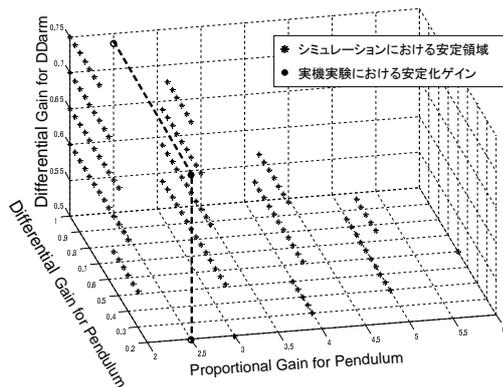


Fig. 6 シミュレーションによる安定化可能ゲイン領域

## 2.4 PD制御器による安定化実験

本節では、実機のDDアーム-倒立振り子系に対し、Fig. 6 に示す領域の周辺でゲインを調節することにより、安定化を試みる。実験ではまずロータリエンコーダによって倒立振り子の角度検出・制御を行い、同時にカメラによる角度検出の基準点調節を行う。その後カメラによる制御に切り替えるという方式をとった。

実験結果をFig. 7 に示す。20秒以前はロータリエンコーダで振り子の角度を検出しているため、むだ時間もなく振り子はほぼ静止している。また、20秒以後はカメラによる角度での制御を行っており、振り子は振動し続けている。この振動は収束も発散もせず、ゲインを調節しても抑えることはできなかった。この実験での視覚情報による制御時のゲインは $(P_p, D_p, P_2, D_2) = (2.5, 0.2, 0.1, 0.73)$ であり、この結果はFig. 6に赤い●印として示した。シミュレーション結果と実験結果が異なるのは、シミュレーションに使用したパラメータと実機のパラメータが異なっていたためと考えられる。また、Fig. 6のシミュレーション結果では安定化できるゲインは様々に存在するように感じられるが、実機における実験では上述したゲイン以外では安定化できなかった。したがって、実機ではシミュレーションよりも安定化できるパラメータ領域はかなり狭いものと思われる。

## 3 JIT手法を用いた倒立振り子の安定化制御

第2節でむだ時間を持つDDアーム-倒立振り子系に対しPD制御器により安定化制御を行った。しかし、人間が手の上で倒立振り子を立てることができるようになるまでに第2節のようにPDゲインを細かく調節しているとは考えにくい。また、むだ時間系の制御の本質である予測も行っていない。予測によりむだ時間を補償する制御法は、スミス

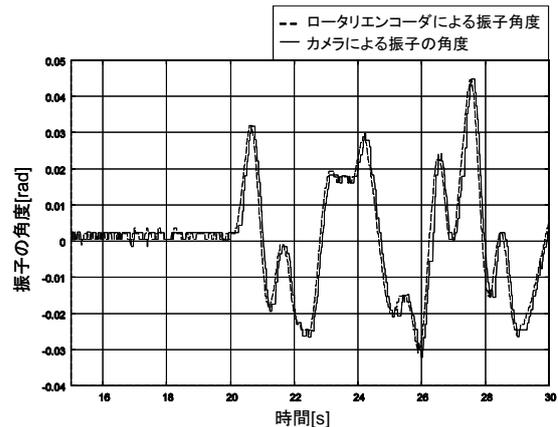


Fig. 7 実機でのPD制御器による安定化実験結果

法、状態予測制御が知られている<sup>7)</sup>。しかし、制御対象のモデルパラメータやむだ時間が正確にわかっている必要があるなど、実機への実装には多くの障害がある。本節では、制御対象の構造やパラメータなどの事前情報を必要としない、人間の学習・記憶のメカニズムをモデリングしていると言えるJIT手法を、むだ時間を含む本システムへ応用する。JIT手法における振り子の振舞いと、人間のStick BalancingにおけるStickの振舞いを比較することにより、本手法が人間の運動制御の再現に対する一つの解釈となりうることを示す。

### 3.1 JITモデリング

非線形システムに対する同定手法のひとつとして、JITモデリングと呼ばれる手法が提案され<sup>8, 9, 10)</sup>、実システムへの適用事例が報告されている<sup>11)</sup>。JITモデリングでは、同定対象の大域的なモデルを求める代わりに、システムの過去の入出力データを大量に蓄積し、蓄積されたデータから局所モデルを構成し、出力の推定値を得る方法である。大域的なモデルを構成する際には、制御対象の構造や物理現象に関する事前情報が必要となるが、JIT手法ではより少ない事前情報のみで同定を行うことができるという利点がある。この特徴を生かすことにより、複雑で大規模な非線形システムや、人の行動や状況判断といった数学モデルでは記述できないシステムの振舞いを同定する問題への応用が期待できる。

JIT手法のもうひとつの特徴として、JIT手法におけるモデルとはシステムの過去の入出力データから構成されるデータベースそのものであることが挙げられる。このことは、過去のシステムの振舞いを記憶し、新しい行動の決定や状況判断を行う人間の運動制御の仕組みを連想させる。実際に、JIT手法によって得られるシステムの出力の推定値には、過去の入出力データのダイナミクスが直接的に反映される。したがって、JITモデリングは、過去のシステムの振舞いを記憶し、それらを

状況に応じて取り出して活用する役割を果たしていると言える。

同定対象から新しいデータが得られるたびに同定モデルを更新するオンライン同定法は、一般に Eager Learning と呼ばれている。これに対して、JIT モデリングでは、新たに得られたデータに対しては蓄積のみを行い、同定の必要性が生じた場合に限り、次に述べる手順により蓄積データを利用して局所モデルを構成する。このような同定法は Lazy Learning と呼ばれている<sup>12)</sup>。

次式で表される離散時間非線形システムを考える。

$$y(t) = f(\psi(t)), \quad t = 0, 1, 2, \dots \quad (5)$$

ここで、 $y(t)$  は式(5)のシステムの出力、 $f(\cdot)$  は非線形写像、 $\psi(t)$  は情報ベクトルと呼ばれている。情報ベクトル  $\psi(t)$  は、システムの過去の入出力データを用いて次式で定義される。

$$\psi(t) := \begin{bmatrix} y(t-1) & \dots & y(t-n_y) \\ u(t) & u(t-1) & \dots & u(t-n_u) \end{bmatrix}^T \quad (6)$$

ここで、 $u(t)$  はシステムの入力であり、 $n_y$ 、 $n_u$  はそれぞれシステムの出力と入力の次数であり、我々が自由に選ぶことができる。JIT モデリングでは、(6)式の情報ベクトルの形式でデータの蓄積が行われる。

時刻  $t$  において、(5)式のシステムの出力  $y(t)$  の予測を JIT 手法により行うこと考える。このときの  $\psi(t)$  は要求点 (Query, 以後この  $\psi(t)$  を  $q$  で表す) と呼ばれ、この要求点に類似した情報ベクトルを蓄積データから選び出し、近傍として採用する。情報ベクトルの類似度は、情報ベクトル間の距離  $d$  を Euclid 距離の近さによって測る。

$$d(\psi(t), \psi(i)) := \sqrt{(\psi(t) - \psi(i))^T (\psi(t) - \psi(i))} \quad (7)$$

近傍の最も単純な選び方は、要求点からもっとも近い  $k$  個の情報ベクトルを集める方法であり、 $k$ -NN ( $k$  Nearest Neighbors) と呼ばれている<sup>12)</sup>。 $k$ -NN 以外にも、同定精度を向上させるために  $k$ -BN ( $k$  Bipartite Neighbors)、 $k$ -BN2、 $k$ -SN ( $k$  Surrounding Neighbors)、 $k$ -SBN ( $k$  Surrounding Bipartite Neighbors) という様々なバリエーションが提案されている<sup>8, 9)</sup>。

要求点  $q$  に対する近傍が得られたとき、局所モデルを構成する代表的な方法として、重み付き局所回帰法 (LWR) と重み付き線形平均法 (LWA) が挙げられる。

$$\text{LWR} : \begin{cases} \hat{y}(t; q) = \psi^T(t; q) \hat{\theta} \\ \hat{\theta} := \arg \min_{\theta} \sum_{i=1}^k w_i |y(i) - \hat{y}(i)|^2 \end{cases} \quad (8)$$

$$\text{LWA} : \hat{y}(t; q) = \frac{\sum_{i=1}^k w_i y(i)}{\sum_{i=1}^k w_i} \quad (9)$$

ただし、 $\hat{y}(t; q)$  は局所モデルによる出力の推定値であり、 $k$  は近傍のサイズである。

JIT モデリングの重要なポイントは、近傍の決定は要求点  $q$  との距離のみによって行われるため、データがいつ蓄積されたかということには依存しないということである。さらに、各要求点  $q$  毎に局所モデルが構成されるため、 $q$  に対する出力の予測後は局所モデルはその都度破棄されることが、従来の同定法と大きく異なる。

## 3.2 JIT 手法を用いた安定化実験

以下の手順で JIT 手法による安定化制御を実現する。

### • Step 1: 振舞いの記憶

ロータリエンコーダによる角度検出を行い、安定可制御を行う。ロータリエンコーダによるむだ時間を含まない振子フィードバック制御入力  $u_\phi$  と、むだ時間を含むカメラによる角度検出  $\phi_{\text{img}}$  から情報ベクトルを構成し、データベースに格納する。つまり、式(5)、(6)において

$$u_\phi(t) = f(\psi(t)), \quad t = 0, 1, 2, \dots \quad (10)$$

$$\psi(t) := \begin{bmatrix} \phi_{\text{img}}(t) & \dots & \phi_{\text{img}}(t - n_\phi) \end{bmatrix}^T \quad (11)$$

とした。十分なデータが確保されるまで、Step 1 を繰り返す。制御系の概略を Fig. 8 に示す。実験では、 $n_\phi = 3$  とした。

### • Step 2: 記憶の再現

カメラによる角度検出を JIT 手法における要求点として、JIT 手法を用いて速度指令値の予測値を得る。実験では近傍数  $k = 4$  とし、LWA を採用した。また、式(9)における重みは、できるだけ制御入力  $\hat{y}(t; q)$  を滑らかにするために、次式で与える。

$$w_i = \exp \left\{ - \frac{(y(i) - \hat{y}(t-1; q_{t-1}))^2}{2} \right\} \quad (12)$$

一周期前の推定値  $\hat{y}(t-1; q_{t-1})$  の近傍で大きくなるような重みを選んでいる。Step 2 の制御系の概略を Fig. 9 に示す。

実験結果を Fig. 10, 11 に示す。実験では Step 1 は 40 秒間行い、その後 Step 2 に切り替えた。Fig. 10 は Step 1: 振舞いの記憶の結果である。振子を様々な振幅や周期で動かすことにより、それに対応した制御入力をデータベースに蓄積している。実際に加えているのは、ロータリエンコーダによる PD 制御入力であるが、同時に JIT 手法による制御入力も表示した。JIT 手法により推定された制御入力がある程度確認できる。Fig. 11 は、Step 2: 記憶の再現の結果である。実際に、JIT 制御入力により制御を行っている。参照用に PD 制御入力も同時に示した。大まかではあるが推定が上手く行われ、むだ時間のない理想的な制御入力に近い値となることが確認できる。振子の鉛直上向きへの制御は 16 秒間達成された。わずか 40 秒の記憶・学習で 16 秒間も安定化できたことにより、人間の何らかの運動制御と比較しても、JIT 手法が上手く機能していると言える。また、制御不可となったのは、JIT 手法の特徴である学習していない状況には対応できないことが原因であるが、この特徴はまさ

に人間らしいと言える．Step 1をより長く効率的に行うことによって，より多くの状況を学習し長時間の安定化制御も可能である．

## 4 まとめ

カメラ-倒立振り子系という不安定なむだ時間をもつ系に対して，単純なPD制御器と，人の学習・記憶のメカニズムをモデリングしていると言えるJIT推定器により安定化を試みた．どちらも漸近安定化は達成されていないが，PD制御器では収束も発散もしない特定の周期に陥ることを示した．また，JIT手法はむだ時間システムの同定と振子の安定化制御を同時に行うアプローチである．この手法はPD制御器と異なり，PDゲインを調節する必要がない．この考え方は，人間の記憶とそれに基づいた意思決定の機構を工学的な枠組みで具体化しているという点で興味深い．

## 参考文献

- 1) J. L. Cabrera and J. G. Milton, On-Off Intermittency in a Human Balancing Task, *Physical Review Letters*, Vol. 89, 158702 (2002)
- 2) J. L. Cabrera and J. G. Milton, Stick Balancing: On-Off Intermittency and Survival Times, *Nonlinear Studies*, Vol. 11, No. 3, 305/317 (2004)
- 3) J. L. Cabrera and J. G. Milton, Human Stick Balancing: Tuning Lévy Flights to Improve Balance Control, *Chaos*, Vol. 14, No. 3, 691/698 (2004) *Nonlinear Studies*, Vol. 11, No. 3, 305/317 (2004)
- 4) 牛田 俊, 福田 憲一郎, 李 眞娥, 出口 光一郎, 人の学習運動制御系の振舞いに基づく倒立振子のむだ時間補償制御, 第48回自動制御連合講演会講演論文集, 長野, (2005)
- 5) 李 眞娥, 牛田 俊, 福田 憲一郎, 出口 光一郎, むだ時間を含むセンサ情報による倒立振子の安定化制御, 計測自動制御学会東北支部第226回研究集会, 資料番号226-7, (2005)
- 6) 出口 光一郎, ロボットビジョンの基礎, コロナ社, (2000)
- 7) 渡部 慶二, むだ時間システムの制御, コロナ社, (1993)
- 8) 鄭 秋宝, 木村 英紀, Just In Timeモデリングの新しい手法とその圧延セットアップモデルへの応用, 計測自動制御学会論文集, Vol. 37, No. 7, (2001)
- 9) Q. Zheng and H. Kimura, Just-In-Time Modeling for Function Prediction and its Applications, *Asian J. Control*, Vol. 3, No. 1, 35/44 (2001)
- 10) 牛田 俊, 木村 英紀, Just In Timeモデリング技術を用いた非線形システムの同定と制御, 計測と制御, Vol. 44, No. 2, (2005)
- 11) 伊藤 雅浩, 松崎 眞六, 大貝 春俊, 大館 尚記, 内田 健康, 斎藤 信一, 佐々木 望, 高炉 操業における大規模データベースオンラインモデリング, 鉄と鋼, Vol. 90, No. 11, (2004)
- 12) D. W. Aha: *Lazy Learning*, Boston, London: Kluwer Academic Publishers (1997)

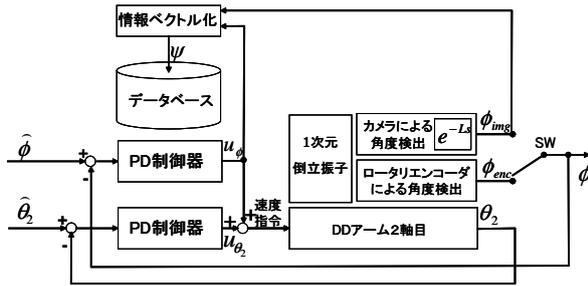


Fig. 8 振舞いの記憶

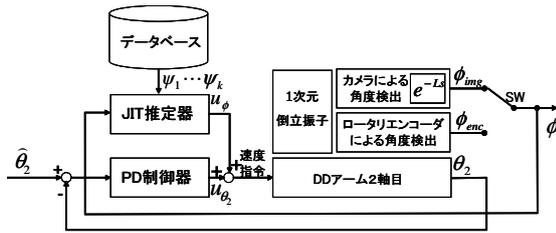


Fig. 9 記憶の再現

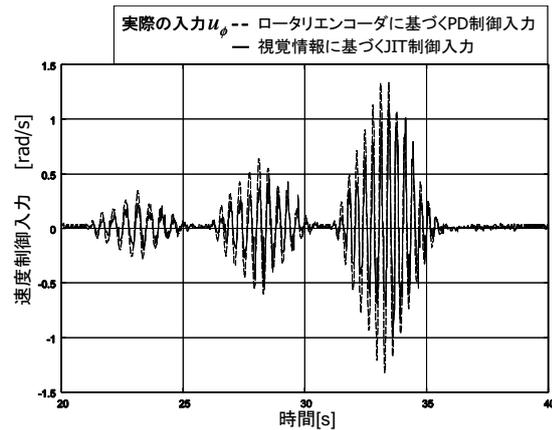


Fig. 10 振舞いの記憶の結果

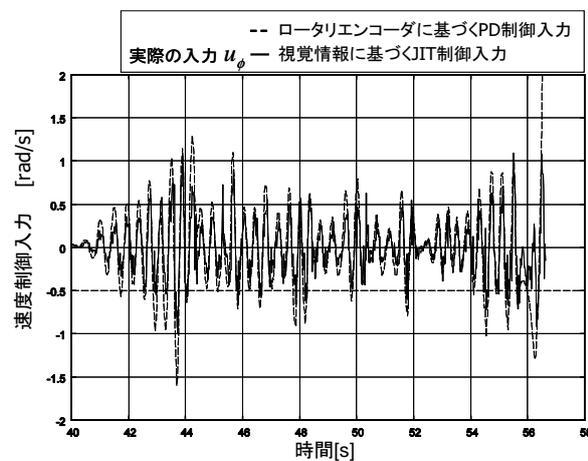


Fig. 11 記憶の再現の結果