論　文

感覚運動統合システムにおけるダイナミックス整合の適応的獲得

尾川　順子[†a)]　　阪口　豊[††]　　並木　明夫[†††,†]　　石川　正俊[†]

Adaptive Acquisition of Dynamics Matching in Sensory-Motor Fusion System

Naoko OGAWA[†a)], Yutaka SAKAGUCHI[††], Akio NAMIKI[†††,†], and Masatoshi ISHIKAWA[†]

あらまし　感覚運動統合システムにおける設計コンセプトとして「ダイナミックス整合」が提案されている．これは，物理的・計算的制約のもとでシステムの時間特性を調節してパフォーマンスの最大化を目指す，という概念である．本論文ではダイナミックス整合問題を最適化問題としてモデル化し，その適応的獲得アルゴリズムを構築する．また，アクティブビジョンによるターゲットトラッキングタスクを例題とした数値実験によって，本手法により合理的な解が獲得されることを示す．

キーワード　感覚運動統合，ダイナミックス整合，強化学習，ターゲットトラッキング，ロボティクス

1. まえがき

近年のロボティクスにおいては要素技術の発展が目覚ましく，高速に動作するセンサやアクチュエータの開発により，従来では考えられなかったような超高速でタスクを実行する能力が期待されている [1]〜[3]．しかしながら，システムの動作が高速になるにつれて，各要素の様々な時間特性（ダイナミックス）の影響が無視できなくなってくる．特に，アクチュエータやセンサの能力による物理的な時間特性と計算資源やアルゴリズムなどによる計算上での時間特性は，システムのパフォーマンスに大きく影響するようになる．

例えば，サーボモータの制御には一般に少なくとも 1 kHz の制御周期が必要とされているため [4]，センシング情報をアクチュエータの制御に用いるには，それに見合ったサンプリング周波数をもつセンサを導入することが要求される．逆に，高速なセンシングシステムを導入しても，アクチュエータが遅ければセンサの高速性を十分に活用できない．更にシステムのパフォーマンスは，処理系で用いられるアルゴリズムや情報処理戦略にも大きく依存する．

すなわち，ロボットの感覚系，処理系，運動系を矛盾なく統合し，感覚運動統合システム [5] としてそのパフォーマンスを最大限に発揮させるには，各要素単独での高速性のみを議論しても無意味であり，互いの時間特性を考慮し，整合させることが必要となる．

このような観点から，Namiki らは物理的・計算的な時間特性を統一的に扱う枠組みとして，dynamics matching（ダイナミックス整合）という概念を提唱した [5], [6]．これは，「物理的，計算的な制約のもとで，システムの感覚系，処理系，運動系それぞれの時間特性（ダイナミックス）を調整し，タスクや外界のダイナミックスに整合させることで，状況に応じたパフォーマンスの最大化を目指す」というものである．

例えば，センサの精度やアクチュエータの出力などの物理的な能力の不足したロボットを，予測や推定などの計算により補償することを考える．しかし，プロセッサの演算速度や記憶装置の大きさなどから，実時間での計算量には一定の限界がある．したがって，1）物理的な能力不足を補償するために計算量を増やす必要性，2）実時間性を維持するために計算量を抑える必要性，という二つの相反する要請があり，ここから生じるトレードオフのもとで，情報処理戦略や計算量

† 東京大学大学院情報理工学系研究科，東京都
Graduate School of Information Science and Technology, Univ. of Tokyo, 7–3–1 Hongo, Bunkyo-ku, Tokyo, 113–8656 Japan
†† 電気通信大学大学院情報システム学研究科，調布市
Graduate School of Information Systems, Univ. of Electro-Communications, 1–5–1 Chofugaoka, Chofu-shi, 182–8585 Japan
††† 科学技術振興機構 CREST，川口市
CREST, Japan Science and Technology Agency, 4–1–8 Honcho, Kawaguchi-shi, 332–0012 Japan
a) E-mail: Naoko_Ogawa@ipc.i.u-tokyo.ac.jp

などをどうダイナミカルに制御するか，という問題が生じる．ダイナミックス整合の実現は，システムの物理的時間特性と計算的時間特性を制御して，このような問題を解決する過程にほかならない．

従来の研究では，このような物理的・計算的時間特性の統一的扱いが十分とはいえず，また現場の技術者の経験則や勘に頼る場合が多かった．ロボティクス分野では，古くから軌道計画などにおいてロボットの物理的なダイナミックスを扱ってきたが [7]〜[9]，計算資源や計算負荷などについては陽に議論されてこなかった．その結果，ロボットのハードウェアやソフトウェアがもつ潜在能力を生かしきれていなかった．逆に人工知能や経営工学の分野では，一定の計算資源のもとでの推論や意思決定が研究されており [10], [11]，また並列処理やハードリアルタイムスケジューリングなどの分野では資源割当が重要な話題である [12], [13]．しかし多くは物理的ダイナミックスを考慮しておらず，実世界のロボットシステムに適用するには不十分であった．ダイナミックス整合の概念は，システム設計におけるこの問題の解決を模索する一つの試みである．

さて，ダイナミックス整合の実現には，以下に述べるような二つのレベルが考えられる．一つは，システム設計段階における実現というレベルであるが [5], [6]，各要素の特性を事前に把握するのは一般には困難であり，特性が変動する場合への適用も難しい．もう一つは，状況の変動に応じてダイナミックス整合の状態をオンラインで適応的に獲得させるというレベルであり，探索的手法を用いれば，特性が事前に分からない状況にも適用可能である．

本論文では，後者のような適応的なダイナミックス整合を最適化問題とみなしてモデル化し，適応的獲得アルゴリズムを提案する．またターゲットトラッキングタスクを例題として，このアルゴリズムを数値実験により実装する．

2. ダイナミックス整合の適応的獲得アルゴリズム

2.1 問題設定

まずシステム，タスク，外界全体によって規定される，系全体の種々の特性のダイナミックスに着目する．これらの特性には物理的なもの（例：モータの最大トルク，可動範囲，センサのダイナミックレンジ，センサの感度など）もあれば，計算的なもの（例：計算資源，アルゴリズム，意思決定など）もある．またこれらの特性のダイナミックスは，システム自身にとって直接的に調整できないものと，直接調整できるものとに分類できる．以下，前者を「制約ダイナミックス」$\boldsymbol{c}$，後者を「可変ダイナミックス」$\boldsymbol{a}$ と呼ぶことにする．また，システムパフォーマンスを P で表す．このときダイナミックス整合問題は「与えられた制約ダイナミックス $\boldsymbol{c}$ のもとで，パフォーマンス P を最大化するような最適な可変ダイナミックス $\boldsymbol{a}$ を見つける」という最適化問題とみなせる．

図 1 はこの最適化問題を図示したものである．図の底面はシステムの特性を成分とするベクトルによって張られる空間であり，その軸は制約ダイナミックス $\boldsymbol{c}$ と可変ダイナミックス $\boldsymbol{a}$ とに分かれている（簡単のため，両者の相互作用はないと仮定する）．縦軸はパフォーマンスの分布 $P(\boldsymbol{a}, \boldsymbol{c})$ を表し，一般にはシステムにとって未知である．ここで，ある制約ダイナミックス $\boldsymbol{c}'$ が与えられるということは，$\boldsymbol{c} = \boldsymbol{c}'$ という固定された超平面でこの空間を切断することになり，その断面である部分空間は制約ダイナミックス $\boldsymbol{c}'$ 下でのパフォーマンスの分布 $P(\boldsymbol{a}|\boldsymbol{c}')$ を表している．我々の目標は，この部分空間内で可変ダイナミックス $\boldsymbol{a}$ を動かし，パフォーマンス P が最大になるような $\boldsymbol{a}^*$，すなわち，

$$\boldsymbol{a}^* = \operatorname*{argmax}_{\boldsymbol{a}} P(\boldsymbol{a}|\boldsymbol{c}')$$

を探すこととなる．

この最適化問題は，1) 未知である $P(\boldsymbol{a}|\boldsymbol{c}')$ を適応的に推定する，2) $P(\boldsymbol{a}|\boldsymbol{c}')$ を最大化する $\boldsymbol{a}$ を見つける，という二つの要素から構成されている．これらを同時

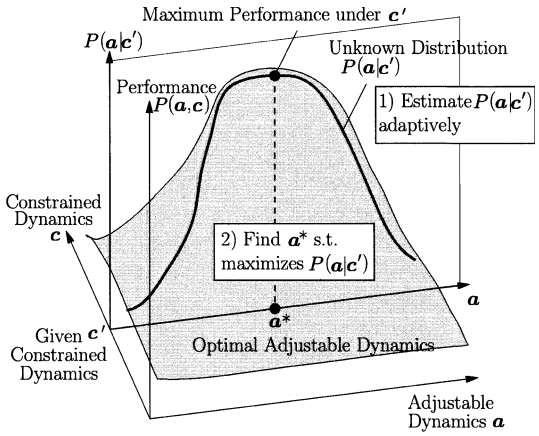

図 1　ダイナミックス整合問題のモデル
Fig. 1　A model of the dynamics matching problem.

に解決することは解析的手法では困難であり，探索的な解法が必要である．また，扱う対象がダイナミックスである以上，モデル化が不可能な部分も多く，実機を動かしながらオンラインで解を求めることが望ましい．そのため並列的な探索はほぼ不可能であり，可変ダイナミックスの値ごとに逐次的に試行することが要求される．これらの要請に対して，今回は実世界との相互作用に基づく試行錯誤的な手法としてロボティクスと親和性のある，強化学習を用いてこの探索を実現することとした．強化学習は，携帯電話の動的チャネル割当 [14] やファジー制御のパラメータの調整 [15] など大規模で複雑な問題の解決手段として優れた実績がある．次節では，ダイナミックス整合問題を強化学習問題として定式化し，これを解決するアルゴリズムを提案する．

2.2 強化学習に基づく獲得アルゴリズム

強化学習は，環境から得られる報酬の最大化を目標とする試行錯誤的な学習方法である [16]．獲得すべき目標パターンが明示的に与えられなくても学習が進行するという点で，教師あり学習とは大きく異なる．

強化学習の学習主体はエージェントと呼ばれる．エージェントは環境の「状態」を観測し，ある「政策」に従って「行動」を起こす．環境はそれによって状態遷移するとともに，行動の善しあしを表す量である「報酬」をエージェントに与える．この報酬の期待値である「価値関数」が最大になるように学習が進む．

報酬の期待値を最大にするには，報酬の予測を行う必要がある．エージェントは環境との相互作用の中で，価値関数を推定していくことで状態や行動を評価し，報酬を予測する．エージェントはこの価値関数をもとに政策を定め，行動を選択していく．

このような特徴をもつ強化学習を，以下では図 2 に示すように，ダイナミックス整合の獲得問題に適用する．まず，超平面 $\boldsymbol{c}=\boldsymbol{c}'$ で切断された断面，$P(\boldsymbol{a}|\boldsymbol{c}')$ に注目する．この超断面の底面は，可変ダイナミックス $\boldsymbol{a}$ を元とする $(\dim \boldsymbol{a})$ 次元空間である（簡単のため図 2 では 1 次元として描いている）．この空間を $\mathcal{P}$ とする．

提案するアルゴリズムでは，空間 $\mathcal{P}$ 上の各点 $\boldsymbol{a}$（可変ダイナミックス）を強化学習における「状態」とみなし，空間 $\mathcal{P}$ 上での移動 $\Delta\boldsymbol{a}$ を「行動」とみなすことにする．そして，即時的なパフォーマンスの指標となるような何らかの量を「報酬」に対応づける．このとき価値関数 $V(\boldsymbol{a},\Delta\boldsymbol{a})$ の分布は，学習が進むにつれてパフォーマンス $P(\boldsymbol{a}|\boldsymbol{c}')$ を色濃く反映していくようになる．

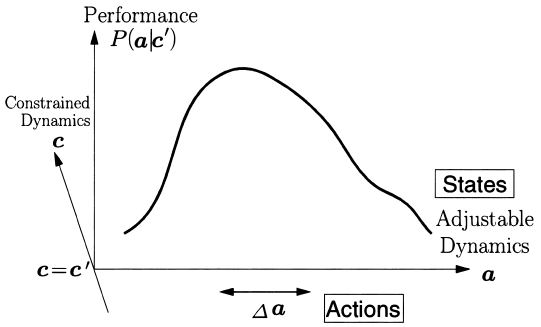

図 2　強化学習のダイナミックス整合問題への適用
Fig. 2　Application of reinforcement learning to the dynamics matching problem.

これによりアルゴリズムの目的は，可変ダイナミックスの調節により状態空間上を試行錯誤的に動きながら，タスク全体のパフォーマンスを最大化するような状態を見つけ出すこととなる．

なお，以降は簡単のため，状態空間 $\mathcal{P}$ を離散空間と仮定するが，連続空間に対しても拡張可能である．また，今回はパラメータ空間の探索を gridworld 問題 [16] のアナロジーととらえることにより，行動を可変ダイナミックスの変化量として定義した．しかし，行動の定義はダイナミックス整合にとっては本質的な問題ではないため，任意性があり，適用する問題に応じた定義が可能である．

以上の仮定をもとに，提案する獲得アルゴリズムは以下のようなものとする：

（1） 現在自分が選択している可変ダイナミックス $\boldsymbol{a}$ を把握する．（状態観測）

（2） 価値関数 $V(\boldsymbol{a},\Delta\boldsymbol{a})$ を参照し，可変ダイナミックス $\boldsymbol{a}$ の変化量 $\Delta\boldsymbol{a}$ を決める．（行動選択）

（3）（2）で選ばれた変化量 $\Delta\boldsymbol{a}$ に従い，可変ダイナミックス $\boldsymbol{a}$ が変化する．（状態遷移）

（4） 新しい可変ダイナミックス $\boldsymbol{a}'$ を把握する．（遷移先の状態観測）

（5） 新しい可変ダイナミックスのもとで何らかのタスクを行い，報酬を獲得する．（報酬獲得）

（6） 報酬の善しあしをもとに価値関数 V を評価し，更新する．（価値関数更新）

（7）（1）に戻る．

3. ターゲットトラッキングタスク

本章では，以上のようなダイナミックス整合の獲得アルゴリズムの有用性を検証する一つの例題として，アクティブビジョンによるターゲットトラッキングタスクを取り上げる．

システム設計論において具体性と一般性のバランスをとることは簡単ではなく，実機に即した例題では実機特有の問題に引きずられるため，一般的な議論が難しくなる．本論文では簡略化された仮想的なシステムを例題とすることで，一般的観点から検証を行う．

3.1 アクティブビジョンとダイナミックス整合

アクティブビジョンは，移動装置を備えた視覚システムである．近年，様々な分野で視覚認識の重要性が増すにつれて，アクティブビジョンにより高い処理能力が要求されるようになってきている．これに対し，視覚処理の高速化など，一部の性能はハードウェア上で改善されつつあるが [17]，アクチュエータの速度やトルク，センサの解像度やサイズなどの多くの制約のために，単純なアルゴリズムではパフォーマンスを上げるのは困難である．その一方で，実時間性を維持するため，載せられるアルゴリズムや情報処理戦略にはステップ数の厳しい制約がかかる [18]．ここから，ハードウェア制約と実時間制約に対し，情報処理のダイナミックスをいかに整合させるかという問題が生じる．本節では，アクティブビジョンによるターゲットトラッキングタスクを題材として，この問題を考える．

ターゲットトラッキングタスクとは，運動するターゲットをアクティブビジョンで追跡するタスクである．タスクの概要を図 3 に示す．

ここでは提案手法の有用性を分かりやすく示すために，簡略化された仮想的なシステムを想定した．簡単のためターゲットは 2 次元平面上を運動する点とし，カメラもそれに平行な 2 次元平面上を運動できるとする．観測の時間遅れや観測誤差はここでは問題としない．

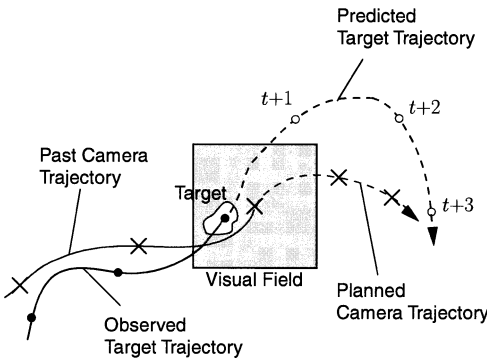

図 3 ターゲットトラッキングタスクの概要
Fig. 3 Outline of a target-tracking task.

システムは観測情報からターゲットの動きの内部モデルを構築し，ターゲット軌道を数ステップ先まで予測する．そして，予測に従ってハードウェア的制約を補償するような軌道計画を行い，実際に運動する．この「観測 → モデル構築 → 予測 → 軌道計画 → 運動」という一連の処理を一定時刻ごとに繰り返すものとする．

3.2 ターゲットトラッキングタスクにおけるダイナミックス整合

本節では，このタスクにおけるダイナミックス整合を考える．**2.1** で導入した制約ダイナミックス $\boldsymbol{c}$ は，ここではターゲットの運動のダイナミックス，カメラの性能やモータの性能の限界及び情報処理の計算負荷として現れてくる．また可変ダイナミックス $\boldsymbol{a}$ としては，情報処理戦略の制御が挙げられる．以下では制約ダイナミックス $\boldsymbol{c}$，可変ダイナミックス $\boldsymbol{a}$，パフォーマンス P を，表 1 のように定義する．ここで可変ダイナミックス $\boldsymbol{a}$ として選ばれている予測段数 k（何ステップ先までの運動を予測するか）は予測に費やす計算量の指標であり，ターゲットの軌道予測とカメラの軌道計画とに計算量をどう割り当てるかという戦略を制御するものである．

ターゲットの軌道予測においては，遠い将来まで予測すれば，その結果を軌道計画に反映して厳しい制約を補償できる．その結果，ターゲットを見失う危険性が小さくなり，システム全体のパフォーマンスを保証できると期待される．しかし，k を上げて遠い将来まで予測するほど処理時間 T_p がかかる．また，カメラ

表 1 ターゲットトラッキングタスクとダイナミックス整合問題の対応付け
Table 1 Correspondence between the target-tracking task and dynamics matching problem.

制約ダイナミックス $\boldsymbol{c}$	ターゲットのダイナミックス c カメラ視野の幅 d [mm] カメラの最大移動速度 v [mm/step] 計算時間の上限 $T = T_\mathrm{p} + T_\mathrm{t} (\leq 1)$ [step]
可変ダイナミックス $\boldsymbol{a}$	予測段数 k
パフォーマンス P	ターゲットの予測誤差の 2 乗平均の逆数 $1/\mathrm{avg}(\lvert\boldsymbol{x}(t) - \boldsymbol{x}_\mathrm{p}(t)\rvert^2)$ [mm^{-2}]

の軌道計画においても，軌道の精度を上げようとするほど，一般に反復計算などにより時間 T_t がかかる．一方，実時間性を維持するためには，処理全体にかけられる時間 $T_\mathrm{p}+T_\mathrm{t}$ は有限となる．そのため，例えばパフォーマンスを向上させようとして予測にかける計算量を増やしすぎると，軌道計画に時間をかけられなくなる．その結果，軌道の精度が低下して，結果的にはターゲットを見失いかねない．

このことから図 4 のように，処理の増大と実時間性との相反によってトレードオフの関係が生じる．このトレードオフの条件は事前には未知であり，制約ダイナミックスの値によって変化するため，予測と軌道計画にそれぞれ割り当てる計算量のバランスをダイナミカルに制御することが必要となる．これを今回は予測段数 k の制御により実装する．

これによりこの問題は，制約ダイナミックス $\boldsymbol{c}$ のもとで予測段数 k をどう選べば平均 2 乗予測誤差が最小になるか，すなわち，

$$k^* = \operatorname*{argmax}_k P(k|c,d,v,T) \tag{1}$$

なる k^* を求める，ということになる．

システムにとって最適な予測段数 k を教えてくれる教師は存在しないため，強化学習による試行錯誤的な解探索が有効と考えられる．特に，オンラインで k を探索するには，各試行で k を変化させて実際に予測誤差を計測する必要がある．次節では **2.2** で提案したアルゴリズムをこの例題に対して適用する．

なお，実際のシステムでパフォーマンスを向上させるには，今回のような単一のパラメータではなく，複数の可変ダイナミックスパラメータを同時に扱う必要があると考えられる．ここで提案する手法は強化学習を用いているために，パラメータを複数化しても同一のアルゴリズムで対応可能である．ただし，簡単化のためとアルゴリズムの有効性を明確に評価するために，まずは可変ダイナミックスは k のみとして実験を行い，複数パラメータについては **3.7** で検討する．

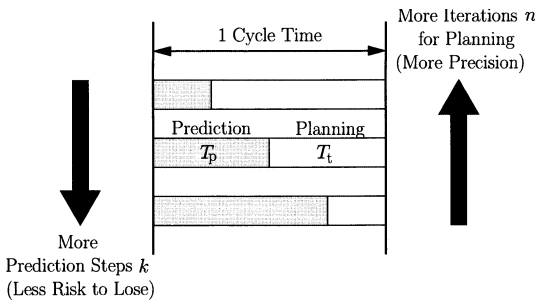

図 4　ターゲットの軌道予測とカメラの軌道計画とのトレードオフ
Fig. 4　Trade-off between the target position prediction and the camera trajectory planning.

3.3　提案するアルゴリズム

2.2 で提案したアルゴリズムに従い，前節の設定を強化学習の枠組みに書き換えると表 2 のようになる．この問題は，各制約ダイナミックスのもとで予測段数を離散的に並べた 1 次元 gridworld を探索する問題と等価となる．

価値関数構築アルゴリズムとしては，代表的な TD 学習である Q 学習 [16] を用いた．Q 学習における価値関数 Q は制約ダイナミックス $\boldsymbol{c}$ をパラメータとしてもち，その更新則は，

$$\begin{aligned}&\Delta Q(s_t, a_t \,|\, \boldsymbol{c}) \\ &\quad = \alpha\Big(r_{t+1} + \gamma \max_a Q(s_{t+1}, a \,|\, \boldsymbol{c}) - Q(s_t, a_t \,|\, \boldsymbol{c})\Big)\end{aligned} \tag{2}$$

で与えた．また，行動選択確率 $\Pr(a)$ の導出には softmax 法 [16]

$$\Pr(a) = \frac{\exp(\beta Q(s_t, a \,|\, \boldsymbol{c}))}{\sum_{a'} \exp(\beta Q(s_t, a' \,|\, \boldsymbol{c}))} \tag{3}$$

を用いた．

このアルゴリズムのブロック図を図 5 に示す．

3.4　実験の設定

以上のようなアルゴリズムに基づいて，ターゲットトラッキングタスクの数値実験を行った．実験の設定は以下のようである．

3.4.1　制約ダイナミックス

まずターゲットの運動 c については，長径の大きさが 200 mm から 400 mm までの 3 種類の楕円（短径は 100 mm で一定），及び擬似乱数を用いたブラウン運動を用意した．カメラのハードウェアについては，視野

表 2　強化学習の各変数の記述
Table 2　Description of each variable in reinforcement learning.

状態 s_t	ある時刻 t にシステムが採用している予測段数 k
行動 a	予測段数 k を 1 段増やす 予測段数 k を 1 段減らす 予測段数 k を維持する
報酬 r_t	（N ステップでの平均 2 乗予測誤差+1）の逆数 $\times 100$

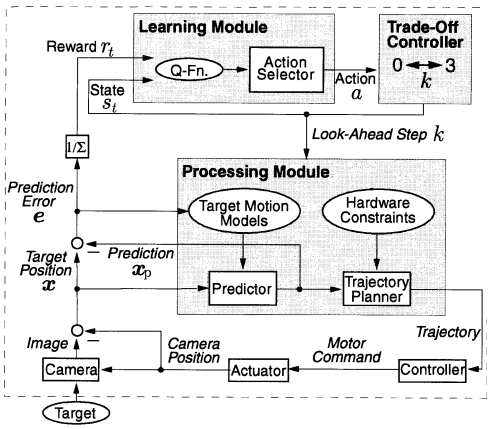

図 5　提案するアルゴリズムの構造
Fig. 5　Structure of the proposed algorithm.

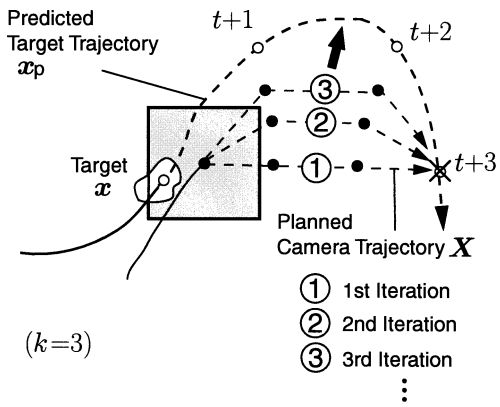

図 6　カメラの軌道計画
Fig. 6　Camera trajectory planning.

の一辺の長さ d を 250 mm から 500 mm まで 6 種類，最大移動速度 v を 100 mm/step から 300 mm/step まで 9 種類用意した．これらの制約ダイナミックスは，試行ごとにランダムに切り換わるものとした．処理時間の制約ダイナミックス T は今回は一定値 1 とした．

3.4.2　可変ダイナミックス

予測段数 k は 0，1，2，3 の 4 種類を選択できるとした．ここで $k=0$ とは予測を行わず，現在見えているターゲットの位置を目標とすることを指す．この場合，予測誤差は追従誤差で代用した．

3.4.3　ターゲットの予測

フィードフォワード制御によってターゲットをトラッキングするには，まずターゲットの運動 $\boldsymbol{x}(t)$ の内部モデルの構築が必要となる．ここではターゲットの種類 c に応じた内部モデルを個別に保持できるものとし，それぞれは 2 次の状態空間表現 $\{A, \boldsymbol{B}\}$ を用いて，ターゲットの 2 次元位置 $\boldsymbol{x}(t)$ を $\boldsymbol{x}(t+1) = A\boldsymbol{x}(t) + \boldsymbol{B}$ という形で表すものとした．また，内部モデル自体は学習初期には不明確であるとし（真の値に白色雑音を乗せた），予測誤差を用いた最急降下法によって，トラッキングと同時並行的に漸次更新した．予測値 $\boldsymbol{x}_{\rm p}(t+1)$ はこのモデルから計算されるとした．予測段数 k の場合には，これを逐次適用することで $\boldsymbol{x}_{\rm p}(t+k)$ を求めることができる．

3.4.4　軌 道 計 画

カメラには移動速度の制限があるので，予測されたターゲットの軌道に沿ってはトラッキングできない可能性があり，カメラの軌道計画を適切に行う必要がある．一般にロボットの軌道計画では最適制御が用いられることが多く，システムが非線形系であることから，反復近似計算が必要となる．以下の軌道計画アルゴリズムは，従来研究における反復近似計算を抽象化したものである．軌道計画は図 6 に示すように，逐次的に精度を上げていく方法を用いた．カメラ軌道の初期値は $\boldsymbol{x}_{\rm p}(t+k)$ への直線軌道 $\boldsymbol{X}(t+i)(i=1,\ldots,k)$ とし，最大移動速度 v の範囲内で，この直線軌道を最急降下法

$$\Delta\boldsymbol{X}(t+i) = \mu(\boldsymbol{x}_{\rm p}(t+i) - \boldsymbol{X}(t+i)) \quad \text{for } i = 1,\ldots,k$$

によりターゲットの軌道へ反復的に近づけていった（μ は正定数．一般に，安定性の点から小さな正数が望ましいとされている）．ここで，ダイナミックス整合の概念により，予測及び軌道計画の計算時間を陽に考える必要がある．処理にかけられる時間は一定値 T に制限されているため，先読み量 k が多くなるほど予測のための処理時間は増大し，逆に軌道計画にかけられる時間は減少する．ここでは簡単のため，この軌道計画の反復計算の回数 n を k に比例した回数 νk（ν：適当な正数）だけ減らすというペナルティを設けることで，この制約を実装した．

なお，ターゲットを途中で見失ってしまった場合には探索モードに移行し，カメラを一定速度でランダムに移動させてターゲットを探すようにした．この間は予測や軌道計画は行わず，ターゲットが視野に入ったら，探索モードから抜けて再びトラッキングを始めるものとした．

3.4.5　学習パラメータ

Q 学習などの種々のパラメータは，$\alpha = 0.01$,

$\beta = 0.05\times$（経過エピソード数），$\gamma = 0.999$，$\mu = 0.1$，$n = 9$，$\nu = 2$，$N = 30$ とした．

3.5 実験結果

3.5.1 制約の厳しさと予測段数の関係

図 7 (a) に示すのは，視野角 d と最大速度 v をそれぞれ変化させたとき，最終 2,000 エピソードにおいて選択された予測段数の平均値の分布である．各ブロックの色の濃さはそれぞれの制約ダイナミックスにおける予測段数の多さを表している．(b) は同様に，ターゲット軌道 c を変化させたときの予測段数の平均値である（上段横軸が楕円の長径）．

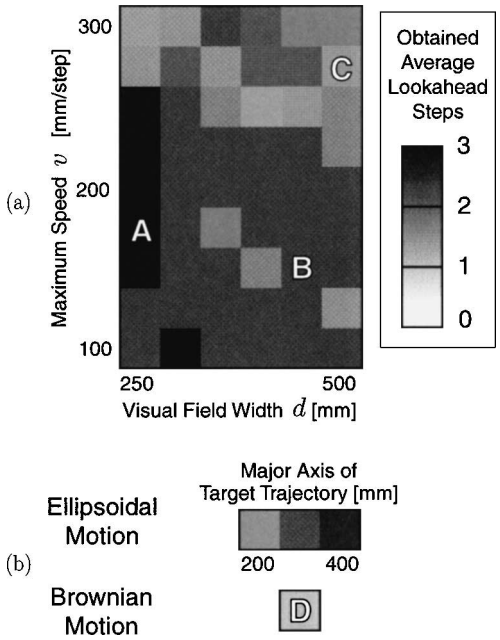

図 7　最終的に獲得された予測段数の平均値の分布図．(a) 感覚・運動系の制約，(b) 外界の制約

Fig. 7　Distribution map of averages of obtained lookahead steps: (a) under sensory-motor constraints, (b) under external constraints.

図 7 から，制約が厳しい（すなわち視野が狭い，カメラの最大移動速度が小さい，ターゲットの速度が大きい，など）場合ほど予測段数が多くなる傾向にあることが分かる．これは，以下のように解釈できる．まず制約条件が緩い場合には，視野中心にターゲットを据えれば，次の時刻でターゲットを見失うリスクは最小となる．今回の軌道計画の設定では，予測・軌道計画の段数を 1 段にすることでこれを達成でき，それ以上の予測・軌道計画は計算負荷のペナルティによりかえってパフォーマンス低下を招く可能性がある．システムは強化学習を通じてこのことを試行錯誤的に「発見」し，予測段数を低く維持したと考えられる．一方，制約が厳しい場合にはこのような戦略ではターゲットをすぐに見失いかねない．そこで，システムは予測・軌道計画の段数を増やすことで制約を補償し，計算負荷を上げてでも数時刻先までのパフォーマンスを保証するという戦略をとったと考えられる．更にターゲットがブラウン運動の場合には，もはや予測に意味がないことをシステムが「発見」し，予測段数を低くしたと考えられる．

なお，図 7 では，獲得された予測段数の分布は必ずしも一様ではない．原因として，今回の実験のような厳しい条件下ではパラメータのわずかな相違にパフォーマンスが影響されやすいことなどが考えられる．これらの改善は今後の課題である．

3.5.2 予測段数の収束の様子

図 8 は，図 7 内の A，B，C，D 各点において，状態（予測段数）が推移していく様子を示している．各点において予測段数がほぼ一定値に収束していく様子が見てとれる．D 点におけるプロットは他の 3 点と比べて収束に時間がかかっており，値も不安定である．この理由として，ターゲットの運動が毎回異なり，そ

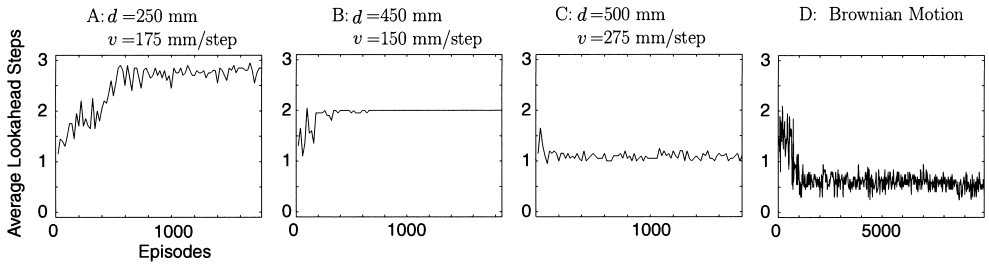

図 8　予測段数の推移（20 エピソードごとの平均値）

Fig. 8　Changes in lookahead steps (average values per 20 episodes).

れによって最適な予測段数も 0 段と 1 段との間で変動するためと考えられる．

3. 5. 3 パフォーマンスの向上

図 9 は，k を適応的に変化させた場合（提案手法）と k をあらかじめ 0，1，2 及び 3 に固定した場合（各 k を維持する行動のみを行わせた場合）とのそれぞれにおいて，価値関数 Q の推移を比較したものである（$d = 450\,\mathrm{mm}$，$v = 150\,\mathrm{mm/step}$．提案手法については $k = 2$ を維持する行動に対する価値関数を示す）．この場合，$k = 2$ 固定が最適であることが見てとれるが，提案手法ではシステムがそれを「発見」し，最終的に $k = 2$ 固定の価値関数に追い付くことに成功している．価値関数はパフォーマンスを大きく反映しているといえるので，本手法は高パフォーマンスな情報処理戦略の獲得に有用であるといえる．

また図 10 は，図 7 内の B 点における予測の平均 2 乗平方根誤差の推移をプロットしたものであり，誤差の減少が確認された（誤差に上限があるように見えるのは数値実験の性質によるものであり，初期にターゲットを見失ってタイムアウトした場合に相当する）．なお，その他の点においてもほぼ同様に誤差が減少していくことが確認された．これらの結果により，本手法が実際にパフォーマンス向上を達成できたといえる．

3. 5. 4 振舞いに見る情報処理戦略の違い

情報処理戦略の違いは，振舞いというマクロなレベルにも現れている．図 11 は，物理制約が厳しい場合（$d = 300\,\mathrm{mm}$，$v = 250\,\mathrm{mm}$）と緩い場合（$d = 300\,\mathrm{mm}$，$v = 400\,\mathrm{mm}$）それぞれで観察されたトラッキングの様子の典型例である．制約が緩いときは予測段数を 1 にしてターゲットを視野中心に据え，ターゲットの軌道にほぼ沿いながらトラッキングしている．一方，制約が厳しいときには，予測段数を 2 に

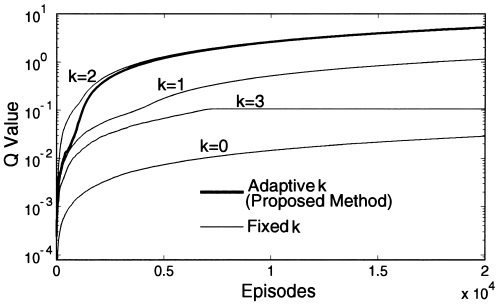

図 9 k を適応的に変化させた場合（提案手法）と固定の場合とでの価値関数の推移の比較
Fig. 9 Comparison of performance with adaptive k (proposed method) and fixed ks.

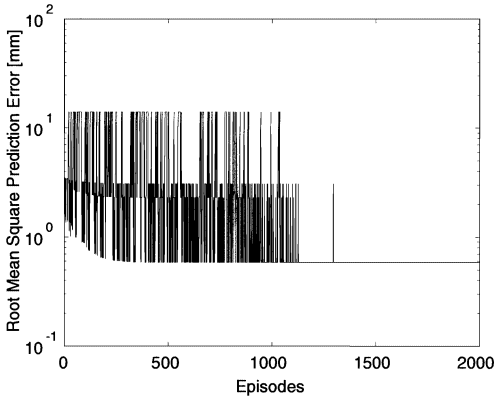

図 10 予測の平均 2 乗平方根誤差の推移の例
Fig. 10 An example of changes in root mean square prediction error.

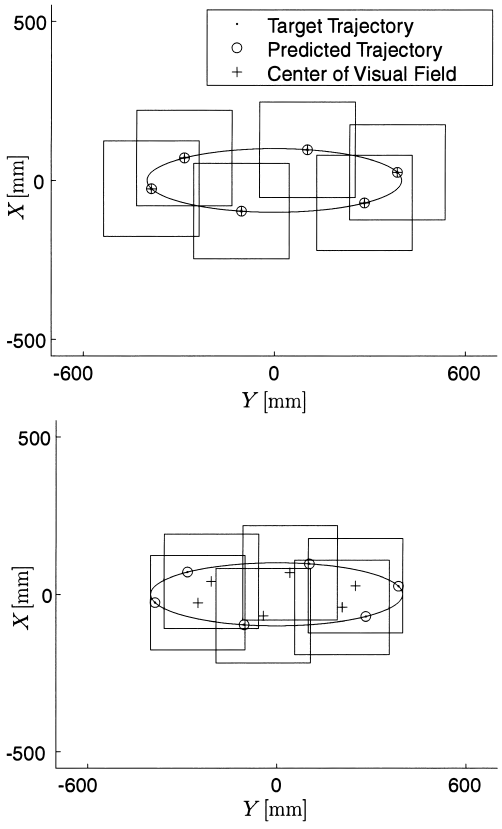

図 11 トラッキングの様子．上段：制約が緩いとき（予測段数 1），下段：制約が厳しいとき（予測段数 2）
Fig. 11 Behavior of tracking. Upper: under loose constraints (1-step lookahead), lower: under severe constraints (2-step lookahead).

して画面の端を有効利用することにより，カメラをあまり動かさずに視野全体をフルに使ってトラッキングする，という合理的な戦略を獲得できている．これは，アクチュエータの制約が厳しいアクティブビジョンに適したトラッキング戦略といえる．

3.6 環境の変動に対する適応性

オンライン学習による適応的手法の有用性を示すため，変動する環境に対しても実験を行った．具体的には，制約ダイナミックスの一つであるターゲット軌道 c を実験中に突然変化させ，システムがそれに応じて最適な予測段数を切り換えられるかどうかを実験した．システムはターゲット軌道の変動を検出し（ここでは検出方法は考察の対象としないが，様々な手法が提案されている [19]〜[24]），softmax 法の逆温度パラメータ β を初期値にリセットするものとした（時間経過により指数関数的に低下していた学習能力を復帰させるためのものであり，学習結果自体は一切リセットされないことに注意）．なお本手法では，以前の学習内容は新しい内容によって上書きされる．以前の内容を保存することも技術的には可能と考えられるが，本論文では考察の対象としない．

5,000 エピソード経過後，ターゲット軌道の楕円長径が 350 mm から 250 mm に変化した場合の，予測段数の推移を図 12 に示す．軌道変化前は予測段数を 3 にしていたが，軌道変化後は予測段数を 2 に切り換えていることが分かる．軌道変化前の方が制約が厳しいことを考慮すると，システムは変動する環境に応じたダイナミックス整合を実現しているといえる．

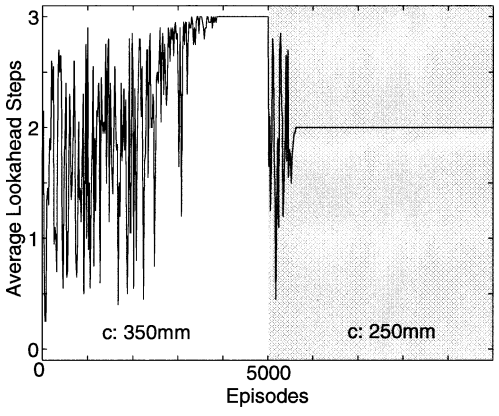

図 12 ターゲット軌道 c が変化する環境での予測段数の推移
Fig. 12 Changes in lookahead steps according to the shift in the target trajectory.

3.7 より複雑な問題に対する適用可能性

以上の実験では，アルゴリズムの有効性を明確に示すために，1 次元 gridworld 問題と等価な設定を用いたが，実システムにおけるダイナミックス整合問題は，より高次元で複雑な場合も多い．本論文で示したアルゴリズムは，原理的には高次元問題にもそのまま適用可能である．しかし次元数が増加するほど，一般に学習時間は増大する傾向にある．以下では学習時間の観点から，より複雑な問題に対する適用可能性を考える．

学習には確率的要素が多分に含まれるため，学習時間の厳密な評価はそもそも不可能である．しかし，学習時間をごく粗く見積もることは可能である．Whitehead は，Q 学習の学習時間が状態空間のサイズに対して指数関数的に増大することを示した [25]．

前節までの実験で考慮された要素は，図 4 にあるように，ターゲットの軌道予測の処理時間 T_p とカメラの軌道計画の処理時間 T_t の二つであった．ここに更に，ターゲット内部モデルの教師あり学習の処理時間 T_s という要素を加えた，三つの要素からなる課題を考えた．具体的には，ターゲット内部モデル獲得の最急降下法の反復回数をもう一つの可変ダイナミックス k' とし，この値に応じても軌道計画の反復回数を減らすペナルティを設けた．この問題は k と k' をパラメータとする 2 次元 gridworld 問題と等価となり，価値関数は 2 次元分が増えて 4 次元となる．実験では，最急降下法反復回数 k' は 1，15 の 2 種類，または 1，5，10，15 の 4 種類とし，$v = 150\,\mathrm{mm/step}$，$d = 450\,\mathrm{mm}$，$c = 300\,\mathrm{mm}$ で固定とした．選択された k と k' の両方が 200 エピソードにわたって一定値になれば学習完了とした．

学習完了までに費やしたエピソード数の 10 回の試行での平均を表 3 に示す．問題が複雑になるに従い，確かに学習時間が増大することが確かめられた．また，価値関数の要素数が 144 と大きい場合，20,000 エピ

表 3 学習完了までのエピソード数．図中 “(4×3)×(2×3)” は，k が 4 状態，3 行動，k' が 2 状態，3 行動であったことを示す
Table 3 The number of episodes spent for learning. The expression “(4×3)×(2×3)” means that k has four states and three actions, and k' has two states and three actions.

可変ダイナミックス数	価値関数の要素数	episode 数
1（k のみ．control）	$4 \times 3 = 12$	1,432
2（k 及び k'）	$(4 \times 3) \times (2 \times 3) = 72$	2,616
2（k 及び k'）	$(4 \times 3) \times (4 \times 3) = 144$	4,148

ソードが経過しても学習が収束しない例がわずかに見られた（表 3 の集計からは除外した）．更に，要素数が 72 の場合にはモデル学習よりトラッキングを優先する明確な戦略が獲得されたが，要素数が 144 の場合には獲得された戦略にはっきりしたパターンは見られず，トレードオフの関係が三つ巴の複雑な状況であったことがうかがえる．

実機は一般に自由度が少なく，可変ダイナミックスの数も少ない場合が多いため，求解不可能になることはそれほど多くないと思われるが，高自由度では学習時間の爆発的増加により，本手法の適用に限界が生じる．実際の設計では，学習時間をあらかじめ見込んだ上で，獲得するべき自由度の数を定める必要がある．また学習時間をできるだけ抑えるには，様々な高速化アルゴリズムの利用も効果的であろう [25], [26]．

また，次元が高くなると価値関数自身に割くべきリソースも増加するが，近年の記憶装置容量の発展により，ダイナミックス整合実現においてはそれほど問題ではなくなってきている．価値関数のリソース量はタスクに大きく依存するため，一概に議論できないが，圧縮や特徴量表現などの手法により，価値関数のリソースを抑える手法も多く提案されており [27], [28]，必要に応じて適用することも有効と思われる．

なお，以上の実験結果では平均 2 乗予測誤差の逆数をパフォーマンスとして定義したが，平均 2 乗トラッキング誤差の逆数を用いた場合でもほぼ同様の結果が得られたことを記しておく．

これらの結果より，提案手法の有用性が明らかになった．ロボットなどの感覚運動統合システムの高速化においてダイナミックス整合の実現は不可欠であるが，今回の例題で検証したようなトレードオフの関係はタスクに依存しない本質的なものである．したがって本手法はそれらに対する汎用的で有効な解決策となると考えられる．

4. むすび

本論文ではダイナミックス整合問題を最適化問題としてモデル化し，その適応的獲得アルゴリズムを強化学習によって構築した．そしてアクティブビジョンによるターゲットトラッキングタスクに応用し，状況に応じて合理的なトラッキング戦略が獲得されることを示すことで，本手法がダイナミックス整合の獲得に有用であることを確認した．

今後はアルゴリズムの収束性の向上などを図るとともに，実機のアクティブビジョン [17] に実装してその有用性を検証する計画である．

文　献

[1] M. Ishikawa, K. Ogawa, T. Komuro, and I. Ishii, “A CMOS vision chip with SIMD processing element array for 1 ms image processing,” Dig. Tech. Papers of 1999 IEEE Int. Solid-State Circuit Conf. (ISSCC’99), pp.206–207, 1999.

[2] M. Kaneko, T. Tsuji, and M. Ishikawa, “The robot that can capture a moving object in a blink,” Proc. 2002 IEEE Int. Conf. Robotics and Automation (ICRA’02), pp.3643–3648, May 2002.

[3] J.A. Guertin and W.T. Townsend, “Teleoperator slave — WAM design methodology,” Industrial Robot, vol.26, no.3, pp.167–177, 1999.

[4] 日本ロボット学会（編），ロボット工学ハンドブック，コロナ社，1990.

[5] A. Namiki, Y. Nakabo, I. Ishii, and M. Ishikawa, “1-ms sensory-motor fusion system,” IEEE/ASME Trans. Mech., vol.5, no.3, pp.244–252, 2000.

[6] A. Namiki, T. Komuro, and M. Ishikawa, “High-speed sensory-motor fusion based on dynamics matching,” Proc. IEEE, vol.90, no.7, pp.1178–1187, July 2002.

[7] M.T. Mason and J.K. Salisbury, Robot hands and the mechanics of manipulation, MIT Press, 1985.

[8] M.W. Spong and M. Vidyasagar, Robot dynamics and control, John Wiley & Sons, 1989.

[9] J.M. Hollerbach, “Dynamic scaling of manipulator trajectories,” in Natural Computation, ed. W. Richards, pp.455–464, MIT Press, Cambridge, MA., 1988.

[10] 椹木哲夫，“モデリングにおける時間と環境の文脈と不確実性，”日本ロボット学会誌，vol.18, no.3, pp.318–324, 2000.

[11] L.R. Beach and T.R. Mitchell, “A contingency model for the selection of decision strategies,” Academy of Management Rev., vol.3, pp.439–449, July 1978.

[12] R. Nigam and C.S.G. Lee, “A multiprocessor-based controller for the control of mechanical manipulators,” IEEE J. Robot. Autom., vol.RA-1, no.4, pp.173–182, Dec. 1985.

[13] G.C. Butazzo, Hard real-time computing systems — Predictable scheduling algorithm and applications, Kluwer Academic, 1997.

[14] S. Singh and D. Bertsekas, “Reinforcement learning for dynamic channel allocation in cellular telephone systems,” Advances in Neural Information Processing Systems 9 (NIPS*96), eds. M.C. Mozer, M.I. Jordan, and T. Petsche, pp.974–980, MIT Press, 1997.

[15] L.O. Hall and M.A. Pokorny, “Averaged reward reinforcement learning applied to fuzzy rule tuning,” Proc. Int. Conf. Fuzzy Logic and Applications, 1997.

[16] R.S. Sutton and A.G. Barto, Reinforcement Learn-

ing: An Introduction, The MIT Press, 1998.
[17] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno, “1 ms column parallel vision system and its application of high speed target tracking,” Proc. 2000 IEEE Int. Conf. Robotics & Automation (ICRA2000), pp.650–655, April 2000.
[18] 石井　抱，石川正俊，“1 ms ビジュアルフィードバックシステムのための高速対象追跡アルゴリズム，” 日本ロボット学会誌，vol.17, no.2, pp.195–201, March 1999.
[19] Y. Sakaguchi and M. Takano, “Learning to switch behaviors for different environments: A computational model for incremental modular learning,” Proc. 2001 Int. Symp. Nonlinear Theory and its Applications (NOLTA2001), pp.383–386, Oct. 2001.
[20] D.M. Wolpert and M. Kawato, “Multiple paired forward and inverse models for motor control,” Neural Netw., vol.11, pp.1317–1329, 1998.
[21] M. Haruno, D.M. Wolpert, and M. Kawato, “MOSAIC model for sensorimotor learning and control,” Neural Comput., vol.13, pp.2201–2220, Oct. 2001.
[22] J. Tani and S. Nolfi, “Learning to perceive the world as articulated: An approach for hierarchical learning in sensory-motor systems,” Neural Netw., vol.12, no.7–8, pp.1131–1141, Oct. 1999.
[23] S. Ishii, W. Yoshida, and J. Yoshimoto, “Control of exploitation-exploration meta-parameter in reinforcement learning,” Neural Netw., vol.15, no.4–6, pp.665–687, June/July 2002.
[24] 水野純也，村越一支，“神経修飾物質系に対応づけた強化学習パラメータの制御法，” 信学技報，NC2002-102, Dec. 2002.
[25] S.D. Whitehead, “A complexity analysis of cooperative mechanisms in reinforcement learning,” Proc. 9th National Conf. Artificial Intelligence (AAAI-91), vol.2, pp.607–613, 1991.
[26] S. Vijayakumar and S. Schaal, “Fast and efficient incremental learning for high-dimensional movement systems,” Proc. 2000 IEEE Int. Conf. Robotics and Automation (ICRA2000), pp.1894–1899, April 2000.
[27] J. Shewchuk and T. Dean, “Towards learning time-varying functions with high input dimensionality,” Proc. 5th IEEE Int. Symp. Intelligent Control, pp.383–388, 1990.
[28] R.S. Sutton and S.D. Whitehead, “Online learning with random representations,” Proc. 10th Int. Conf. Machine Learning, pp.314–321, 1993.

（平成 15 年 4 月 7 日受付，12 月 1 日再受付）

尾川　順子　（学生員）

平 12 東大・工・計数卒．平 14 同大大学院・工・計数修士了．同年，同大学院・情報理・システム情報博士課程進学，現在に至る．日本学術振興会特別研究員．センサフュージョン，学習理論，微生物の運動制御の研究に従事．IEEE，日本バーチャルリアリティ学会，日本ロボット学会各学生会員．

阪口　　豊　（正員）

昭 61 東大・工・計数卒．昭 63 同大大学院・工・計数修士了．同年同大助手．平 6 同大講師．同年電通大院・情報システム学研究科助教授，現在に至る．主に，感覚・知覚及び運動制御メカニズムに関する研究に従事．博士（工学）．日本神経回路学会，日本神経科学学会，日本視覚学会，認知科学会，SFN，ARVO，IEEE 各会員．

並木　明夫

平 6 東大・工・計数卒．平 8 同大大学院・工・計数修士了．平 11 同大学院・工・計数博士了．同年日本学術振興会研究員．平 12 科学技術振興事業団（現・科学技術振興機構）研究員，平 16 東大院・情報理工・システム情報講師，現在に至る．センサフュージョン，知能ロボット，多指ハンドの制御の研究に従事．博士（工学）．日本ロボット学会，日本機械学会各会員．

石川　正俊　（正員）

昭 52 東大・工・計数卒．昭 54 同大大学院修士了．同年，通産省工業技術院製品科学研究所に入所．平元東大・工・計数助教授，平 11 東大・工・計数教授，平 13 東大・情報理工・システム情報教授．現在に至る．生体の情報処理機構の回路モデル，超並列・超高速ビジョン，光コンピューティング，センサフュージョン等の研究に従事．工博．昭 59 計測自動制御学会論文賞，昭 63 工業技術院長賞，平元応用物理学会光学論文賞，平 10 高度自動化技術振興賞（本賞），平 10，13 日本ロボット学会論文賞，平 11 日本機械学会ロボティクス・メカトロニクス部門学術業績賞，平 11 櫻井健二郎氏記念賞，平 12 LSI IP デザイン・アワード IP 優秀賞，平 14 LSI IP デザイン・アワード IP 賞受賞．