予測と制御による階層的運動学習

春野雅彦
国際電気通信基礎技術研究所

1 はじめに

我々は日常生活において高度に複合的、階層的な運動を獲得して行動している。テニスや野球における個々の動作は要素運動が複雑に組み合わされたものであるし、日常の会話でも文、単語、音素といった階層構造の中で発話を行っている。本講義では主としてこのような階層的運動がいかに獲得されるかについて計算論的に考察する。

階層的、複合的運動制御の学習に対しては 2 つのアプローチが考えられる。 1 つは単一のネットワークで問題全体を処理する方法で、他方は複雑な問題をより簡単な部分問題に分割し、各部分問題には少数の特化したネットワーク (エキスパート) を割り当てるモジュラネットワークによる方法である。前者と比較してモジュラネットワークによる方法は学習の局所性という顕著な特徴を持つ。すなわちモジュラネットワークでは適切に説明出来ない事例に対する学習の影響が適切なエキスパートに限定され、他のエキスパートには影響を及ぼさない。したがって早期の学習が可能となる。この学習の局所性ならびに神経生理学的に得られている機能局在やモジュラリティに関する知見から判断して脳における階層的運動学習のモデルとしてはモジュラネットワークの方が適切であると考えられる。

モジュラネットには適用する問題、部分問題への分割法、モジュールの構造及び使用する学習の手法によって様々な種類が考えられる。 2 章ではモジュラネットワークの例として Mixture of Experts [6] を挙げる。これは回帰モデルの一種でありゲーティングネットと呼ばれるネットワークで入力空間を分割し各部分空間に対してエキスパートネットを学習する。 Mixture of Experts は工学的問題に用いられ成功を収めてはいるが、階層のあらゆる場所でゲーティングネットが入力そのものを分割する構造脳のモデルとして適切ではない。

3 章では Multiple Paired Forward and Inverse Models (MPFIM) [8, 4] という運動制御の為に考案されたモデルの説明を行う。ここではゲーティングネットを用いず各エキスパートの予測の良さのみに基づき部分問題への分割が行われる。一般にモジュラネットではタスクを部分問題へ分割する必要があるがこれは池田先生の講義にある通り隠れ変数を持つ推定問題の一種である。 MPFIM に簡単なマルコフ性のモジュール遷移を導入した 2 階層の場合を考察し EM アルゴリズム [1] による学習結果を示す。モデルから考案される行動実験についても簡単に触れる。 4 章では更に一般の階層的 MPFIM で得られた結果について簡単に紹介する。

2　これまでの研究

Mixture of Experts アーキテクチャ [6] は回帰モデルであり入力ベクトル x と出力ベクトル y の対 (x, y) を受取りそれらの間の関数関係を学習する。ゲーティングネット、n 個のエキスパートネットは共に x を入力として受け取り、ゲーティングネットは i 番目のエキスパートネットの寄与率 g_i を、i 番目のエキスパートネットは予測値 y_i を出力する。ゲーティングネットは各エキスパートネットに対応する n 個の出力を持ちその状態を s_i とすれば g_i は soft-max 関数 (1) で定義される。システム全体の最終的な予測値 y は (1) を用いた線形和 (2) で計算される。

$$g_i = \frac{e^{s_i}}{\sum_{j=1}^{n} e^{s_j}} \tag{1}$$

$$y = \sum_{i=1}^{n} g_i y_i \tag{2}$$

次にゲーティングネット、エキスパートネットにおける学習則を考える。ここでは (3) を目的関数として最急降下法による重みの更新を行う。(3) は各エキスパートの分離を良くする様に導入された便宜的な対数尤度であり y^* は教師信号、σ_i は各エキスパートにガウスノイズを仮定した場合の標準偏差である。

$$\ln L = \ln \sum_{i=1}^{n} g_i e^{-\frac{1}{2\sigma_i^2}|y^* - y_i|^2} \tag{3}$$

ネットワークの学習は通常のバックプロパゲーションを用いて行う。はじめにゲーティングネットについて考える。(1) を用いてゲーティングネットの出力変数 s_i で対数尤度 (3) を偏微分すると (4) が得られる。ここで式中の h_i は (5) に示す i 番目のエキスパートが (x, y^*) を生成する事後確率である。(4) と (5) からゲーティングネットの学習はエキスパートネットの寄与率を事後確率に近付ける方向にすすむことが分かる。

$$\frac{\partial \ln L}{\partial s_i} = h_i - g_i \tag{4}$$

$$h_i = \frac{g_i e^{-\frac{1}{2\sigma_i^2}|y^* - y_i|^2}}{\sum_{j=1}^{n} g_j e^{-\frac{1}{2\sigma_j^2}|y^* - y_j|^2}} \tag{5}$$

同様にエキスパートネットの学習則を導出するためにエキスパートネットの出力変数 y_i で対数尤度 (3) を偏微分すると (6) が得られる。(6) から学習には事後確率 (5) で重み付けされた誤差が利用されることが分かる。

$$\frac{\partial \ln L}{\partial y_i} = h_i \frac{(y^* - y_i)}{\sigma_i^2} \tag{6}$$

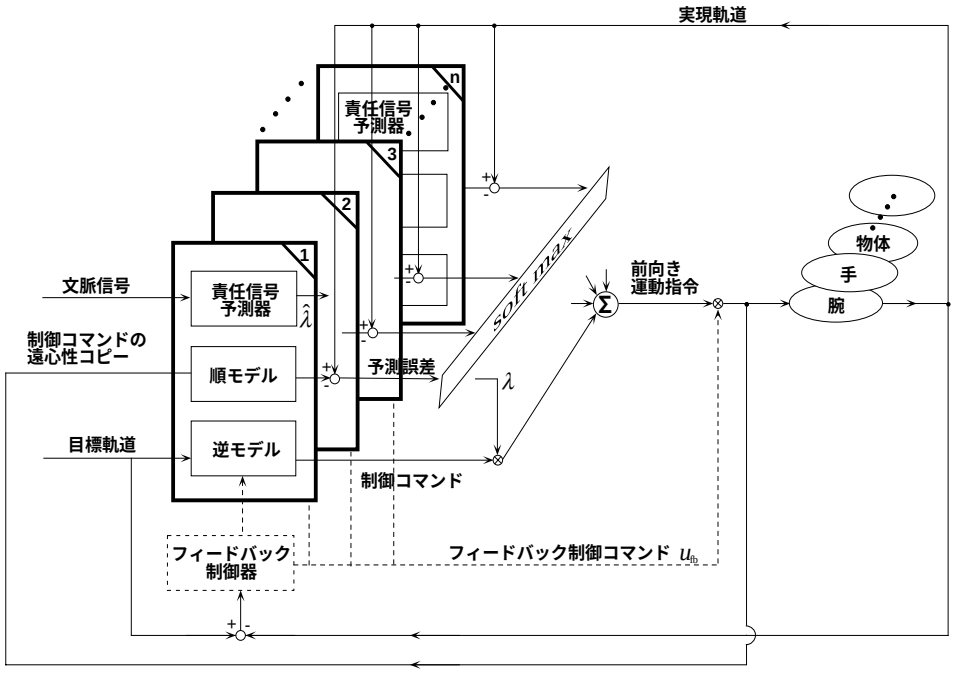

図 1: MPFIM アーキテクチャの概要

3 Multiple Paired Forward-Inverse Models

3.1 MPFIM の概要

本節では Multiple Paired Forward Inverse Model (MPFIM) [8, 4] と呼ばれるアーキテクチャについて説明する。図 1に示す様に MPFIM の各モジュールは目標軌道から制御コマンドを計算する逆モデル、制御コマンドから実現軌道を予測する順モデル、視覚情報等の文脈情報からモジュールの適切さを予測する責任信号予測器の 3 つの部分から構成される。運動制御では各モジュールにおける制御コマンドの誤差を直接知ることは不可能であるため各制御コマンドから実現軌道を予測する順モデルを介する必要が生じる。順モデルの予測と目標軌道を比較することで現在の入力に適切なモジュールを選択し実際の制御及び学習を行うのである。

図 1に示す n 個のモジュールを考える。このとき i 番目の順モデルは (7) の様に現在の状態 x_t、制御コマンド u_t を入力として次時点の状態の推定値 $\hat{x}^i_{t+1}$ を出力とする。ここで w^i_t はニューラルネットワーク ϕ の重みベクトルである。

$$\hat{x}^i_{t+1} = \phi(w^i_t, x_t, u_t) \tag{7}$$

順モデルによる予測を用いて各モジュールの適切さを示す責任信号 λ^i_t を (8) の soft-max 関数で定義する。σ は雑音にガウス分布を仮定した場合の標準偏差である。責任信号は 0 と 1 の間の値を取り現在の状態を良く説明するモジュールほど大きな責任信号値を取る。(9) に示す様に責任信号は順モデルの学習にも利用される。責任信号値の大きなモジュールは多くの誤差を受け取り学習が進展する。責任信号を介した競合学習によってモジュールの分化が進むのである。

$$\lambda_t^i = \frac{e^{-|x_t - \hat{x}_t^i|^2/2\sigma^2}}{\sum_{j=1}^n e^{-|x_t - \hat{x}_t^j|^2/2\sigma^2}} \tag{8}$$

$$\Delta w_t^i = \epsilon \lambda_t^i \frac{d\phi_i}{dw_t^i}(x_t - \hat{x}_t^i) = \epsilon \frac{d\hat{x}_t^i}{dw_t^i} \lambda_t^i (x_t - \hat{x}_t^i) \tag{9}$$

次に逆モデルについて考える。(10) の様に i 番目の逆モデルは目標状態 x_{t+1}^* と現在の状態 x_t を受取り制御コマンド u_t^i を出力する。ただし α_t^i はニューラルネットワーク ψ の重みベクトルである.

$$u_t^i = \psi(\alpha_t^i, x_{t+1}^*, x_t) \tag{10}$$

最終的な制御コマンド u_t は各逆モデルの出力を (11) の様に責任信号で重み付けることで得られる。また逆モデルの学習則は (12) で与えられ、順モデルの場合と全く同様に責任信号を用いて競合的に行われる。ただし逆モデルの場合には真の誤差信号 $u_t^* - u_t^i$ を知ることは出来ないためこれをフィードバック制御コマンド u_{fb} で近似している。

$$u_t = \sum_{i=1}^n \lambda_t^i u_t^i = \sum_{i=1}^n \lambda_t^i \psi(\alpha_t^i, x_{t+1}^*, x_t) \tag{11}$$

$$\Delta \alpha_t^i = \epsilon \lambda_t^i \frac{d\psi_i}{d\alpha_t^i}(u_t^* - u_t^i) = \epsilon \frac{du_t^i}{d\alpha_t^i} \lambda_t^i (u_t^* - u_t^i) \simeq \epsilon \frac{du_t^i}{d\alpha_t^i} \lambda_t^i u_{fb} \tag{12}$$

3.2 MPFIM の適用例

MPFIM を用いた時間的に変化する複数物体の制御シミュレーションについて述べる。図 2に示す通り 5 秒毎に質量、粘性、弾性が α、 β、 γ と規則的に変化する物体を目標軌道 (図 3(b) の最下図) に沿って 30 秒間動かすことが課題である。この問題に対して 3 つのモジュールから成る MPFIM を構成した。このシミュレーションでは各順モデル、逆モデルとして線形ネットワークを用い、 σ は人手により最適値に設定した。線形ネットワークを用いたことで i 番目の順モデル、逆モデルの重みベクトルから M,B,K の推定値 M_i^{F},B_i^{F},K_i^{F} と M_i^{I},B_i^{I},K_i^{I} を計算することが出来る。図 3 (a) は学習に伴う M_i^{F},B_i^{F},K_i^{F} の推移を示しており、ランダムな初期値からはじまって α、 β、 γ へと収束していく。表 1に M_i^{F}、 B_i^{F}、 K_i^{F}、 M_i^{I}、 B_i^{I}、 K_i^{I} の収束値を示すが、これらは α、 β、 γ の良い近似値となっている。表中の全ての場合において順モデルの推定値が逆モデルの推定値よりも優れている。これはまず順モデルから得られる責任信号を用いて逆モデルの学習が制御される MPFIM の特徴を顕著に示している。

モジュール	M_i^{F}	B_i^{F}	K_i^{F}	M_i^{I}	B_i^{I}	K_i^{I}
1	1.0020	2.0080	8.0000	1.0711	2.0080	8.0000
2	5.0071	7.0040	4.0000	5.0102	6.9554	4.0089
3	8.0029	3.0010	0.9999	7.8675	3.0467	0.9527

表 1: 順モデル、逆モデルで獲得された物体の物理特性

(b)

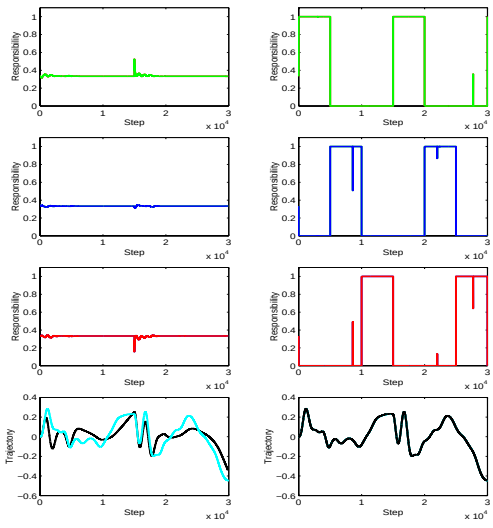

図 3: (a) ランダム値からはじめた 3 物体に対応する順モデルパラメータの学習. (b) 3 つのモジュールの責任信号と運動精度 (最下). 左は学習初期、右は学習終了時.

図 3(b) に学習初期 (左側) と学習終了後 (右側) の各モジュールの責任信号 (上 3 つ) と実現軌道 (最下) を示した。学習初期には各モジュールの分化が進んでおらず目標軌道と実現軌道の差が大きい。それに伴い各モジュールの責任信号も同様の値 (約 1/3) を取っている。これに対して学習終了時には各モジュールの責任信号が順に 1 を取りそれぞれが α、 β、 γ に対応していることが分かる。このときの実現軌道はほぼ目標軌道に一致している。

3.3 マルコフ的にモジュール遷移する MPFIM と EM アルゴリズムによる学習

ここでは Hidden Markov Model (HMM) [5] を用いてマルコフ的にモジュールが遷移する MPFIM を導入し EM アルゴリズムによる学習結果について述べる。上位階層として遷移ダイナミクスを導入した最も簡単な階層モデルと考えられる。 HMM では各モジュール (状態) 間の遷移確率からなる遷移確率行列 P を定義する。 a_{ij} はモジュール i からモジュール j に遷移する確率で時間によらず一定であるものとする。

$$P = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$$

HMM では時刻 t におけるモジュール S_t は N 個のモジュールのうちどれかを取りその間の状態遷移は状態遷移行列 P で規定される。加えて各モジュールにおいて順モデルはそれぞれの出力を行う。i 番目のモジュールの順モデルが持つ線形パラメータを W_i、標準偏差を σ_i とし、入力を X_i とするとこのモジュールが観測データ Y を生成する尤度 $L_i(Y|W_i, \sigma_i)$ はガウスノイズの仮定のもとで以下のようになる。

$$L_i(Y|W_i, \sigma_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} exp^{-\frac{1}{2\sigma_i^2}(Y - X_i W_i)'(Y - X_i W_i)} \tag{13}$$

状態遷移行列も考慮しシステム全体の尤度の期待値を計算すると (14) のようになる。θ は順モデルのパラメータ (W_i,σ_i) を示す。また、ω、s_t はそれぞれ可能なモジュール列の全体、ならびに時刻 t に選択されるモジュールである。

$$L(Y|\theta) = \sum_{\omega} \prod_{t=0}^{T-1} a_{s_{t-1}s_t} L_{s_t}(Y(t)|W_{s_t}, \sigma_{s_t}) \tag{14}$$

詳細は省略するが (14) を最大化するパラメータ P、W_i、σ_i は EM アルゴリズムの 1 種である Baum-Welch アルゴリズム [5] を用いて反復的に解くことが出来る。また逆モデルのパラメータは通常のフィードバック誤差学習によって得られる。

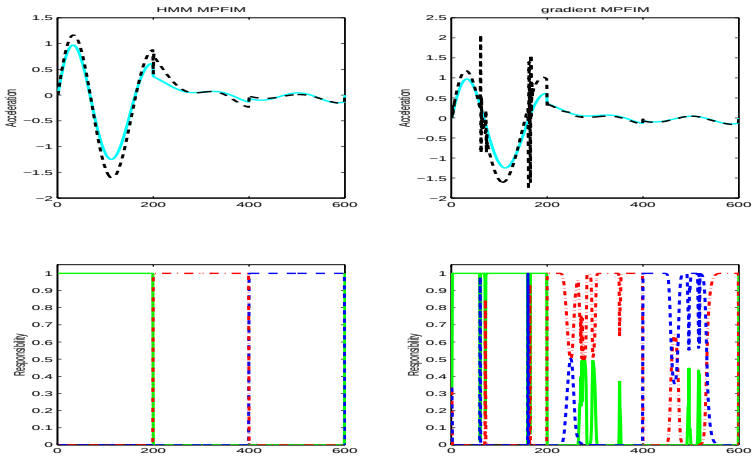

図 4: HMM と通常の学習法における実現された加速度とモジュールの比較

図 4は 3.2 節のシミュレーション課題に HMM (左側) と 3.1 節の式 (9) (右側) を適用した結果を示す [3]。このシミュレーションでは HMM のパラメータ σ_i は学習アルゴリズムにより自動的に推定しているが、式 (9) で用いる σ は人手により最適な一定値に設定している。上側のパネルにおいて実線は目標加速度、点線は実現された

加速度を示し、下側のパネルは各モジュールの責任信号の変化を示す。 HMM ではモジュールの切り替えが適切に行われているのに対し、 (9) の方法では頻繁にモジュールの切り替えが起こっている。これに伴って HMM では加速度の誤差も小さくなっている。これらの差は HMM では状態遷移確率を考慮してモジュールの切り替えを安定化させているために生じていると考えられる。

更に HMM の効果を検討するために様々な切替え数、初期状態で両者の比較を行った。表 2に様々な間隔でモジュールを切り替えた場合の両者の誤差の比を示す。各ネットワークの初期値として真の値を中心に 20%、 40% のノイズを加え 50 回の試行の平均値を示した。この結果から初期値が真の値から遠く、モジュール切替えの頻度が低い場合には HMM の精度が顕著に高くなることが分かる。このようにモジュラネットの学習に EM アルゴリズムを適用することで状態遷移行列 P や競合学習のパラメータ σ_i などを自動的に推定出来るだけでなく、安定して高い精度を得られることが分かる。

Initial conditions s.d.	**Switching period**				
-	5	25	50	100	200
20%	0.998	0.887	1.031	0.636	0.443
40%	0.855	0.676	1.147	0.598	0.322

表 2: 2 種類のノイズに対する加速度誤差の比: HMM-MPFIM/gradient-MPFIM

実際の脳においてもモジュール遷移のダイナミックスのトップダウン情報と実際の誤差に基づく情報の両者のせめぎあいで制御が行われていると考えられる。したがって HMM に基づくモジュール遷移を持つ実験課題を設定し

- 受動的に制御を行う
- 状態遷移は見せて探索的に制御を行う
- 状態遷移も見せずに探索的に制御を行う

を比較することでパラメータの推定、状態の予測に関する脳部位を特定出来るものと考えられる。

4 階層的 MPFIM

これまで述べた MPFIM を更に一般の階層的な場合に拡張した階層的 MPFIM について述べる。図 5(a) に示すとおり各階層は MPFIM から構成されている。一番下位の階層では制御 (逆) モデルが制御コマンドを出力し予測 (順) モデルはそのモジュールの良さを予測する。これに対して上位のモジュールでは予測モデルの役割は同じであるが制御モデルが自分の下の階層の各モジュールに対する事前確率分布を出力する。つまり直接制御コマンドを出すのでは無く抽象的な形で下位のモジュールを制御している。

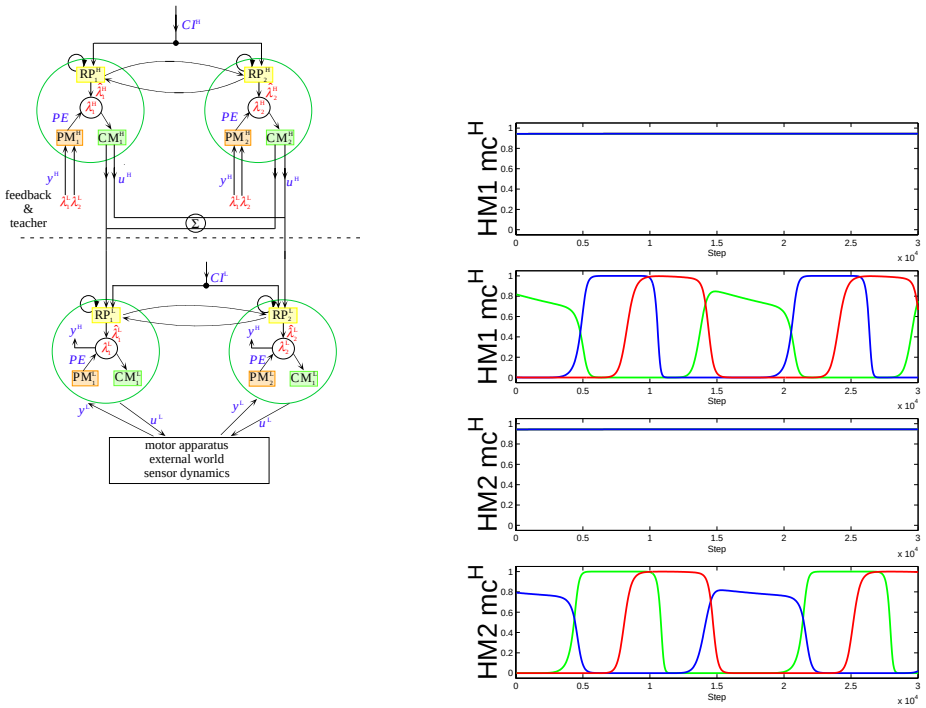

図 5: (a) 一般化された階層的 MPFIM の概念図、 (b) 2 つのシーケンスを学習する前後の上位 2 モジュールの出力。上から学習前、学習後の順に並んでいる

下位モジュールを 3 つ、上位モジュールを 2 つ持つ階層的 MPFIM を用意し、 2.2 節で述べたシミュレーションにおいて 2 種類の系列: 系列 A: $\alpha \rightarrow \beta \rightarrow \gamma$ and sequence 系列 B: $\beta \rightarrow \alpha \rightarrow \gamma$ を学習させた。下位の 3 モジュールは図 3(b) と同様に収束した。上位 2 モジュールの出力を図 5(b) に示す。学習前は全く予測を行えず同じ確率を出力しているが学習後は各モジュールが 2 つの系列に対応して下位モジュールを選択していることが分かる。このように運動系列に固有の反応をするニューロンは猿の SMA や PreSMA でも見つかっており [7]、運動制御の階層性、双方向処理を考える上で興味深い。

5 おわりに

階層的制御モデルとして MPFIM を導入し、学習によって様々な階層性が獲得されることを述べた。高等生物の脳における運動制御ではモジュール的、階層的処理が本質的な役割を果たしており、ここで述べた様な手法は脳の情報処理に関する様々なレベルでのモデル化や実験のデザインに寄与し得る [2]。

参考文献

[1] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. B*, 39:1–38, 1977.

[2] Z. Ghahramani and D.M. Wolpert. Modular decompositioin in visuo-motor learning. *Nature*, 386:392–395, 1997.

[3] M. Haruno, D.M. Wolpert, and M. Kawato. Multiple paired forsward-inverse models for sensorimotor learning and control. *submitted for publication*, 1999.

[4] M. Haruno, D.M. Wolpert, and M. Kawato. Multiple paired forward-inverse models for human motor learning and control. In M. Kearns and S. Solla, editors, *Advances in Neural Information Processing Systems 11*, pages 31–37. MIT Press, 1999.

[5] X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[6] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixture of local experts. *Neural Computation*, 3:79–87, 1991.

[7] J. Tanji and K. Shima. Role for supplementary motor area cells in planning several movements ahead. *Nature*, 371(29):413–416, 1994.

[8] D.M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1998.