神経回路網による冗長マニピュレータの 逆キネマティクス問題の解法[†]

田中 健一*・清水 優子*・土屋 和雄**

A Method of Solution to an Inverse Kinematics Problem of a Redundant Manipulator Using Neural Networks

Ken-ichi TANAKA*, Masako SHIMIZU* and Kazuo TSUCHIYA**

An inverse kinematics problem of robot manipulators with redundancies has a bottleneck of computational time. In this paper, we proposed a method of solution to the above problem using a neural network with sufficient accuracy and short computational time. The learning algorithm proposed here automatically selects a set of joint angles, which maximizes a given cost function using redundant degrees of freedom, from a large number of solutions and forms the inverse kinematics model in the network. Application of the algorithm to the inverse kinematics problems of kinematic singularity and joint limit avoidance is shown as an example.

Key Words: redundant manipulator, neural network, inverse kinematics, inverse problem, kinematic singularity

1. はじめに

FA ロボットに続く次世代ロボットとして自律型ロボットが考えられている。自律型ロボットとは、時々刻々変化する環境の中で自ら判断しながら目的探索行動を行うロボットである。このようなロボットでは特に、環境の変化に対応する柔軟性や即応性が要求されるので、そのマニピュレータには冗長性をもたせることが必要であると考えられる。

冗長マニピュレータはその冗長性を利用して特異点や障害物の回避、対象物の裏側への回り込みなどの動

作ができるので、それをもたないマニピュレータに比べ、操作性、汎用性にすぐれている。そのため、これまでもこの冗長性を利用するためのマニピュレータの制御方法として、ヤコビ行列の零空間への写像を用いて評価関数を最適化する方法^{1),2)}や、特異点近傍で低感度な運動分解を行う方法³⁾などいくつかの報告がなされている。しかし、これまでに提案されている方法は演算量が多いので、FA ロボットのようにあらかじめ決められた軌道に沿う動きをオフラインで計算しておき、それに従ってマニピュレータを動かす場合には適しているが、自律型ロボットのようにその場の状況に応じたマニピュレータのリアルタイム制御が必要な場合にこれらの方法を直接適用することは困難である。このため自律型ロボットの運動制御系として冗長マニピュレータの制御をリアルタイムで行える制御方法が必要となってくる。

このような問題を解決する一つの手段として神経回路網が考えられる。神経回路網ではその処理が並列的に行われるので、応答性が良くなることが期待できる。さらに、実際にマニピュレータを動かしながら、学習によりマニピュレータのモデル誤差をオンラインで補正できることも神経回路網の特徴であり、柔軟な制御方式が実現可能となる。

神経回路網を用いた冗長マニピュレータの制御に関して、堤ら⁴⁾は障害物を回避しながら、ハンドを目標に到達させるための各関節位置を求める方法を提案している。また、中山ら⁵⁾は仮想アームの概念を導入して、障害物回避を考慮したマニピュレータの軌道を与える関節角を求める方法を提案している。これらの方法は、その極小値が問題の解となるようなエネルギー関数を、ハンドや関節の位置、ハンドの目標値などで定義し、それをエネルギーの最小化を行う神経回路網

† 第32回システムと制御研究発表講演会で発表 (1988・5)

* 三菱電機(株)中央研究所 尼崎市塚本町 8-1-1

** 大阪大学工学部 吹田市山田丘 2-1

* Central Research Laboratory, Mitsubishi Electric Corporation, Amagasaki

** Faculty of Engineering, Osaka University, Suita
(Received November 9, 1990)

(Revised February 28, 1991)

を用いて並列的に解くという手法である。ところが、上で述べた方法では神経回路網が平衡状態になるまで繰り返し回路網の状態を更新する必要があるため、演算回数が多いという問題が残されている。

本論文では、冗長マニピュレータの逆キネマティクス問題の解を陽に与える写像関数（逆キネマティクスモデル）を神経回路網に学習させるためのアルゴリズムを提案する。ここで提案するアルゴリズムは冗長自由度を用いて、与えられた評価関数をできる限り大きくするような逆キネマティクスモデルを自動的に回路網内に形成する。この方法により神経回路網を用いて、作業座標から関節角への写像を一度学習させてしまえば、与えられた作業座標に対する関節角をきわめて短時間に得ることができるので、リアルタイム性に優れた制御方式が実現できる。ここでは、この学習アルゴリズムを特異点回避および関節角動作範囲の制限回避に適用した場合を例として取り上げる。

2. 学習アルゴリズム

n 自由度のマニピュレータを考え、その全体の姿勢をベクトル

$$\theta = [\theta_1, \theta_2, \dots, \theta_n]^T \in R^n \quad (1)$$

で表わす。ここで、 θ_i は第 i 関節の関節変数であり、添字 T は転置を表わす。マニピュレータハンドの位置および姿勢を表わすベクトルを

$$\mathbf{r} = [r_1, r_2, \dots, r_m]^T \in R^m \quad (m \leq n) \quad (2)$$

とし、ハンドの位置ベクトルと呼ぶことにする。(1)式と(2)式を用いてハンドと各関節との幾何学的な関係（順キネマティクスモデル）が

$$\mathbf{r} = f(\theta) \quad (3)$$

で与えられる。 $f(\cdot)$ はマニピュレータの形態で決まる関数である。

さて、 m 入力 n 出力の写像を学習する神経回路網を考え、これと(3)式で与えられる順キネマティクスモデルを用いて、Fig. 1 に示すような系を構成する。このような系に適した神経回路網としては、Albus の CMAC^(6,7) や誤差逆伝播法⁽⁸⁾ でしばしば用いられる多層構造の回路網などがある。Fig. 1 において $\mathbf{r}_d$ はハンドの位置ベクトルの目標値、 $\mathbf{r}$ は $\mathbf{r}_d$ を神経回路網に入力して得られる関節変数ベクトル θ で駆動された

マニピュレータハンドの位置ベクトルである。神経回路網を学習させて $\mathbf{r} = \mathbf{r}_d$ になったとする。このとき、写像: $\mathbf{r}_d \rightarrow \mathbf{r}$ は恒等写像となるので、神経回路網の部分には順キネマティクスモデルの逆関数、すなわち逆キネマティクスモデルが得られる。よって、ここで考えるべき問題は $\mathbf{r}$ を $\mathbf{r}_d$ へ近づけるために神経回路網をどのように学習させれば良いかということになる。

(3)式の関係をヤコビ行列 $\mathbf{J}(\theta) \in R^{m \times n}$ を用いて微小変化の式に書き直すと

$$\delta \mathbf{r} = \mathbf{J}(\theta) \delta \theta \quad (4)$$

を得る。ここで $\delta \mathbf{r}$ を、

$$\delta \mathbf{r} = \mathbf{r}_d - \mathbf{r} \quad (5)$$

とおくと、 $\delta \mathbf{r}$ は $\mathbf{r}$ から $\mathbf{r}_d$ へのベクトルであり、この方向へ修正を加えれば、 $\mathbf{r}$ は $\mathbf{r}_d$ へ近づく。したがって、(5)式を(4)式へ代入し $\delta \theta$ について解けば $\mathbf{r}$ を $\mathbf{r}_d$ へ近づけるために必要な回路網の出力 θ における修正量として次式を得る。

$$\delta \theta = \mathbf{J}^+(\theta)(\mathbf{r}_d - \mathbf{r}) + (\mathbf{I}_n - \mathbf{J}^+(\theta)\mathbf{J}(\theta))\mathbf{h} \quad (6)$$

(6)式が(4)式を $\delta \theta$ について解いたときの一般解である。ここで、 $\mathbf{J}^+(\theta)$ はヤコビ行列 $\mathbf{J}(\theta)$ の疑似逆行列、 $\mathbf{I}_n$ は $n \times n$ 単位行列、 $\mathbf{h} \in R^n$ は任意のベクトルである。(6)式の右辺第 2 項が冗長自由度が存在するために生じた項であり、 $\mathbf{h}$ を適当に選ぶことで、冗長性を有効に利用できる。

ところで、 θ で微分可能なあるスカラー関数 $H(\theta)$ を考え、 k を適当な定数として、(6)式の $\mathbf{h}$ を

$$\mathbf{h} = k \nabla_{\theta} H(\theta) \quad (7)$$

ただし、

$$\nabla_{\theta} H(\theta) = \left[\frac{\partial H(\theta)}{\partial \theta_1}, \frac{\partial H(\theta)}{\partial \theta_2}, \dots, \frac{\partial H(\theta)}{\partial \theta_n} \right]^T \quad (8)$$

のように選べば、(6)式の右辺第 2 項が $H(\theta)$ を増加させるように働くことが知られている¹⁾。よって、(7)式を(6)式へ代入した次式にしたがって $\mathbf{r}$ を $\mathbf{r}_d$ へ近づけるように回路網の学習を進めれば、 $\mathbf{r} = \mathbf{r}_d$ を保ちながら冗長自由度を用いて与えられた評価関数 $H(\theta)$ をできる限り大きくするような逆キネマティクスモデルが自動的に得られる。

$$\delta \theta = \mathbf{J}^+(\theta)(\mathbf{r}_d - \mathbf{r}) + k(\mathbf{I}_n - \mathbf{J}^+(\theta)\mathbf{J}(\theta))\nabla_{\theta} H(\theta) \quad (9)$$

Fig. 1 Block diagram of learning

(9)式が学習の基礎式であり、これは入出力間の写像関数を学習する回路網に対して一般的に成り立つ式である。

本論文では、回路網として CMAC^{6),7)}を用いることにする。CMAC は Albus によって提案された小脳をモデル化した神経回路網で、入力層、中間層、出力層の3層からなる。CMAC では入力空間が多く的小領域に分割されており、空間内のすべての点はいずれかの小領域に属している。それぞれ的小領域について一定個数の中間層の連想細胞 (association cell) が結合している。入力空間内で指示点を隣の小領域へ移動すると、一定個数の中から一つだけが別の細胞と入れ替わるように結合がなされている。すなわち、中間層の連想細胞はその受容野を互いに少しずつずらすように入力空間をコーディングしている。その結果、中間層には汎化性が生じる。この入力層と中間層の結合は、学習により変化することなく固定である。

一方、出力層の細胞はすべての中間層の連想細胞と結合しており、その出力は入力空間の指示点で選択された連想細胞との結合の重みを加算して得られる。中間層と出力層の結合の重みは学習により変化するので、逆キネマティクス問題のように入力空間が均質な連続写像関数を学習するのにきわめて適している。以下に(9)式より得られる $\delta\theta$ を用いて表わした、CMAC の結合の重み更新の式を示す。

CMAC は多入力1出力の回路網であるので、 m 入力 n 出力の回路網とするためには CMAC を n 個並列に並べた Fig. 2 のような構成をとる。各関節ごとに一つの CMAC が対応する。第 i 関節に対応する CMAC の出力 θ_i は

$$\theta_i = \sum_{j=1}^M w_j \quad (10)$$

Fig. 2 m -input n -output CMAC

で表わされる。ここで w_j は連想細胞の重み、 M は1回の想起で読み出される重みの個数である。CMACの重みの更新はCMACの出力とそれに対する望ましい出力との誤差で行われるので、(9)式の $\delta\theta$ を直接的に用いることができる。すなわち、 $\delta\theta_i$ を $\delta\theta$ の第 i 番目の成分として、

$$w_j^{NEW} = w_j^{OLD} + \beta \frac{\delta\theta_i}{M} \quad (11)$$

が重み更新の式である。ここで、 β は学習の進み具合を決める定数、添字 NEW および OLD はそれぞれ更新後および更新前の状態を表わす。

つぎに、学習時の注意について述べる。(9)式に示される学習の基礎式は(4)式の微小変化の関係式から導かれている。このため、学習時に δr が過大になると、(4)式が近似的にも満足されなくなり、(9)式を用いて学習を進めることが不可能になる。したがって、 r が r_d に近い値となるような重みを回路網の初期値として設定する必要がある。ところが、マニピュレータの作業空間全域にわたってこのような初期値を設定することは困難である。以上のような理由から、作業空間内の1点を初期値として設定し、その点から始まる連続的な軌道に沿って、ハンドの目標位置ベクトルを小刻みに移動させながら学習を進める必要がある。このようにすると、学習された点の近傍に対しては回路網の汎化性により近似的な解が出力されるので、 δr が過大になりすぎることなく学習を進めることができる。

さらに、このような学習方法を用いると、学習すべき点の個数は多くなるが、隣接する2点の作業座標に対する関節角が不連続になるのを防ぐことができる。

最後に、冗長自由度をもたないマニピュレータの場合 ($m=n$) について少し述べておく。冗長自由度がない場合(9)式の右辺第2項のような零空間に写像されるベクトルを導入できないので、 $r=r_d$ を保ったまま評価関数を考慮した逆キネマティクスモデルを学習させることはできない。しかし、(9)式の右辺第1項だけを用いて、少なくとも $r=r_d$ だけを満足する逆キネマティクスモデルを学習させることは可能である。ただし、 $\det J(\theta)=0$ となる特異点では $J^{-1}(\theta)$ が定義されないで、その近傍では中村らの低感度運動分解法⁹⁾などを学習アルゴリズムに組み込む必要がある。

3. シミュレーション結果

3.1 特異点回避の逆キネマティクスモデル

ここでは平面内を運動する3自由度のマニピュレータ (Fig. 3) に前章の学習アルゴリズムを適用した結

Fig. 3 Three-link manipulator model

果について述べる。マニピュレータの目標軌道は平面内にあるので、このマニピュレータは冗長自由度をもつ。これを用いて特異点を回避する逆キネマティクスモデルを学習させた。神経回路網としては $M=8$ の CMAC を用いた。CMAC に対する入力 $[x, y]^T$ の範囲は

$$\left. \begin{aligned} -(l_1+l_2+l_3) \leq x \leq l_1+l_2+l_3 \\ l_0-(l_1+l_2+l_3) \leq y \leq l_0+l_1+l_2+l_3 \end{aligned} \right\} \quad (12)$$

であり、CMAC が用いる入力の分割数は x, y 共に 200 とした。目標軌道は

$$\mathbf{r}_d = \{[x, y]^T | x=0.274, 0.2 \leq y \leq 1.2\} \quad (13)$$

の直線である。

マニピュレータハンドの初期値 $\mathbf{r}_I$ は

$$\mathbf{r}_I = [0.274, 0.425]^T \quad (14)$$

とした。この初期値を満足する関節ベクトル $\boldsymbol{\theta}_I$ の一つが

$$\boldsymbol{\theta}_I = [85.0^\circ, -170.0^\circ, -10.0^\circ]^T \quad (15)$$

であるので、 $\mathbf{r}_I$ を入力したときの CMAC の出力が $\boldsymbol{\theta}_I$ となるように、該当する連想細胞の重み（各関節の CMAC に対し 8 個ずつ）に $\boldsymbol{\theta}_I$ の各成分の値を等分配して、初期値を設定した (Fig. 4(a))。他の連想細胞の重みの初期値はすべて 0 とした。学習には (13) 式で与えられた直線を等間隔に 200 分割した点を $\mathbf{r}_I \rightarrow \mathbf{r}_A \rightarrow \mathbf{r}_B \rightarrow \mathbf{r}_I$ の順 (Fig. 3) に連続的に一往復して得られる 400 点を一回の学習データセットとして用いた。

冗長性を用いて特異点回避を考慮した逆キネマティクスモデルを学習させるために次式のような評価関数を用いた。

$$H(\boldsymbol{\theta}) = \det \mathbf{J}(\boldsymbol{\theta}) \mathbf{J}^T(\boldsymbol{\theta}) \quad (16)$$

$\sqrt{H(\boldsymbol{\theta})}$ は可操作度²⁾と呼ばれ特異点からの距離を表わす指標である。(16) 式も可操作度も評価関数としてのふるまいは同じであるので、今回はより簡単な (16) 式を用いた。学習の進み具合を決める定数 β は 0.7, k は 20 とした。

シミュレーションの結果を Fig. 4 に示す。Fig. 4 は神経回路網を用いてハンドが目標軌道を追従するように制御したときのマニピュレータのふるまいである。Fig. 4(a) が初期状態、学習が進むにつれて (b), (c) と変化していく。図中 l はデータセットの学習回数を表わす。 $l=2$ では軌道の下の方が目標から大きく離れている。 $l=5$ になるとハンドはほぼ目標軌道に追従している。このときのハンド位置と目標値との rms 誤差は 0.026 m である。学習をさらに続けて誤差が定常になるまでには約 100 回の学習が必要であった (Fig. 4(d))。最終的なハンドの rms 誤差は 6.54×10^{-3} m であり、これはマニピュレータ全長の 0.75% に相当する。

Fig. 5 は特異点回避を考慮しない場合 ($k=0$) である。初期値は $k=20$ の場合と同じであり、約 100 回の学習で収束した。また、このときのハンド位置の rms 誤差は 6.53×10^{-3} m で、これは $k=20$ の場合とほぼ同じである。ところが $k=20$ の場合は特異点を回避するような関節角が得られているのに対し、 $k=0$ の場合は特異点を回避できず、その近傍で関節角速度が大きくなっている。このことは目標軌道に沿ってマニピュレータを動かしたときの可操作度の変化を示した Fig. 6 からわかる。すなわち、 $k=20$ の場合は全領域 ($0.2 \leq y \leq 1.2$) にわたって可操作度が 0.08 以上に保たれているのに対し、 $k=0$ の場合は $0.35 \leq y \leq 0.4$ の付近で可操作度がほぼ 0 になっており、特異点の近くを通過していることがわかる。また、Fig. 4(d) に対応する各関節の動きは連続的な曲線で表わされており (Fig. 7), マニピュレータの制御に適した逆キネマティクスモデルが得られている。

以上の結果から、特異点回避を考慮した逆キネマティクスモデルが (9) 式を用いて学習により回路網内に形成できることが明らかとなった。

3.2 関節角制限のある場合

3.1 節では特異点回避を考慮した逆キネマティクス問題について述べたが、そこでは関節角の動作範囲に制限はないとした。ところが、実際のマニピュレータにはその構造による制約から関節角の動作範囲に制限があるのが普通である。以下では、関節角の動作範囲を満足しつつ、特異点回避を行う逆キネマティクスモ

(a) Initial configuration

(c) Simulation result ($k=20, l=5$)

(b) Simulation result ($k=20, l=2$)

(d) Simulation result ($k=20, l=100$)

Fig. 4

Fig. 5 Simulation result ($k=0, l=100$)

Fig. 6 Measure of manipulability

Fig. 7 Joint angle (without joint angle limit)

デルを得る場合について考える。

そのためにはつぎのような二つの要素からなる評価関数を用いる。

$$H(\theta) = k_1 H_1(\theta) + k_2 H_2(\theta) \quad (17)$$

ただし、

$$\left. \begin{aligned} H_1(\theta) &= \det \mathbf{J}(\theta) \mathbf{J}^T(\theta) \\ H_2(\theta) &= \sum_{i=1}^n g(\theta_i) \end{aligned} \right\} \quad (18)$$

$$g(\theta_i) = \begin{cases} -\frac{1}{2}(\theta_i - \theta_i^{\max})^2 & (\theta_i > \theta_i^{\max}) \\ 0 & (\theta_i^{\max} \geq \theta_i \geq \theta_i^{\min}) \\ -\frac{1}{2}(\theta_i - \theta_i^{\min})^2 & (\theta_i^{\min} \leq \theta_i) \end{cases} \quad (19)$$

$H_1(\theta)$ は(16)式と同じ特異点回避に対する評価関数である。 $H_2(\theta)$ は関節角動作範囲に対する評価関数で、各関節角がその動作範囲から遠ざかるにしたがって小さくなる。ここで、 $\theta_i^{\max}$ 、 $\theta_i^{\min}$ はそれぞれ第 i 関節の動作範囲の上限と下限である。

二つの評価関数に対する重み k_1 と k_2 は、 l を学習回数、(9)式の k は k_1 と k_2 に含まれるとして次式のように設定した。

$$\left. \begin{aligned} k_1 &= 20 \cdot \min \left(1.0, \exp \left(-\frac{l-50}{10} \right) \right) \\ k_2 &= \min \left(0.7, \exp \left(\frac{l}{20} \right) - 1 \right) \end{aligned} \right\} \quad (20)$$

すなわち、学習の前半では $k_1 > k_2$ とし特異点回避を優先的に考慮した逆キネマティクスモデルを学習させる。そして、学習の後半では逆に $k_2 > k_1$ とし動作範

囲の制限を越えている関節を範囲内に引き込むように学習させる。このとき、関節角がすでに動作範囲内に入っている場合、(9)式の右辺は $k_1 \approx 0$ と(19)式より第1項だけが残るので、 $\|\delta\theta\|$ が最小になるように修正量が決定され、ハンドの位置が目標軌道に収束していればその値はほぼ0である。このため評価関数 $H_2(\theta)$ によって、学習の前半で得られた特異点回避を考慮した逆キネマティクスモデルが影響を受けることはなく、動作範囲を越えた部分だけが範囲内に入るように修正される。

3.1 節で得られた逆キネマティクスモデルを用いて目標軌道に沿ってハンドを動かしたときのマニピュレータの各関節は Fig. 7 に示すように

$$\left. \begin{aligned} 43.25^\circ &\geq \theta_1 \geq -26.10^\circ \\ -17.71^\circ &\geq \theta_2 \geq -122.94^\circ \\ -14.92^\circ &\geq \theta_3 \geq -93.23^\circ \end{aligned} \right\} \quad (21)$$

の範囲を動く。そこで、上で述べた評価関数の有効性を確認するために(21)式の範囲内に関節角の制限を設定し、シミュレーションを行った。次式がシミュレーションに用いた各関節の動作範囲である。

$$\left. \begin{aligned} 25.0^\circ &\geq \theta_1 \geq -120.0^\circ \\ 0.0^\circ &\geq \theta_2 \geq -150.0^\circ \\ 90.0^\circ &\geq \theta_3 \geq -150.0^\circ \end{aligned} \right\} \quad (22)$$

すなわち、第1関節の上限に制限を加えた。ところで、回路網の学習が十分に収束したとしてもその出力には誤差が残るので、シミュレーションでは $\theta_i^{\max} - \varepsilon$ と $\theta_i^{\min} + \varepsilon$ をそれぞれ実際の上限と下限として用いた。ちなみに、 ε は 1.5° とした。

Fig. 8 にその結果を示す。図より第1関節の上限

Fig. 8 Joint angle (with joint angle limit)

Fig. 9 Measure of manipulability

が領域 B ($0.45 \leq y \leq 1.02$) で動作範囲の制限に掛かっていることがわかる。このため、それを補償するように第2関節および第3関節が大きく動いている。このときの可操作度の動きを示す Fig. 9 を見ると制限がない場合に比べて領域 B においてわずかに可操作度が低下している。関節角の制限に掛からない領域 A ($0.2 \leq y \leq 0.45$) および領域 C ($1.02 \leq y \leq 1.2$) では動作範囲の制限がない場合と同じ値が保たれている。当然のことながら各関節の動きも領域 A および C では制限がある場合、ない場合とも同じ結果となっている。以上の結果から、関節角動作範囲を満足する特異点回避を考慮した逆キネマティクスモデルが学習により得られたことがわかる。

4. ま と め

本論文では冗長マニピュレータの逆キネマティクスモデルを学習により神経回路網内に形成するための学

習アルゴリズムを提案した。また、このアルゴリズムを特異点回避および関節角動作範囲の制限を考慮した逆キネマティクス問題に適用し、その有効性をシミュレーションにより確認した結果についても述べた。ここで提案したアルゴリズムは逆キネマティクス問題の解法に用いられるだけでなく、一般に、冗長性をもつ順モデルから与えられた評価関数を最大化する逆モデルを自動的に形成するためにも用いることができる。

参 考 文 献

- 1) A. Liégeois: Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms, IEEE Trans. Systems, Man, and Cybernetics, **SMC-7-12**, 868/871 (1977)
- 2) 吉川恒夫: 冗長性を有するロボットの制御, 日本ロボット学会誌, **2-6**, 83/88 (1984)
- 3) 中村, 花房: 関節型ロボットアームの特異点低感度運動分解, 計測自動制御学会論文集, **20-5**, 453/459 (1984)
- 4) K. Tsutsumi and H. Matsumoto: Neural Computation and Learning Strategy for Manipulator Position Control, Proc. of IEEE 1st ICNN, 525/534 (1987)
- 5) 中山, 辻, 伊藤: 仮想アームを用いた冗長マニピュレータの軌道生成法とそのニューラルネットによる実現, 第8回日本ロボット学会学術講演会予稿集, 877/880 (1990)
- 6) J.S. Albus: A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC), J. of Dynamic Systems, Measurement, and Control, Trans. of the ASME, **97-3**, 220/227 (1975)
- 7) J.S. Albus: Data Storage in the Cerebellar Model Articulation Controller (CMAC), J. of Dynamic Systems, Measurement, and Control, Trans. of the ASME, **97-3**, 228/233 (1975)
- 8) D.E. Rumelhart, G.E. Hinton and R.J. Williams: Learning Representations by Back-Propagating Errors, Nature, **323**, 533/536 (1986)