INNC 90 PARIS

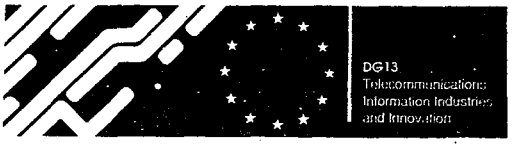
Volume 1

INTERNATIONAL NEURAL NETWORK CONFERENCE

JULY 9-13, 1990
PALAIS DES CONGRES - PARIS - FRANCE

UNDER THE PATRONAGE OF THE COMMISSION OF THE EUROPEAN COMMUNITIES

THE INTERNATIONAL NEURAL NETWORK SOCIETY (INNS), THE IEEE NEURAL NETWORK COUNCIL COOPERATING SOCIETIES



THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

KLUWER ACADEMIC PUBLISHERS
DORDRECHT / BOSTON / LONDON

Proc. Int. Neural Network Conf. INNC-90, Paris 1990

OPTIMAL PERFORMANCE AND STORAGE REQUIREMENTS OF NEIGHBOURHOOD-CONSERVING MAPPINGS FOR ROBOT CONTROL

Dr. R. Brause, J.W.Goethe University, FB Informatik VSFT,
Postbox 111932, D - 6000 Frankfurt 11, West-Germany

Abstract
The new programming paradigm for the control of robot manipulators by *learning* the mapping between the Cartesian space and the joint space (inverse kinematic) is discussed. It is based on a neural network model of optimal mapping between two high-dimensional spaces by Kohonen.
This paper presents the conditions for optimal mappings, based on the principle of maximal information gain. It is shown that Kohonens mapping in the 2-dimensional case is optimal in this sense. Furthermore, the principal control error made by the learned mapping is evaluated for the example of the commonly used PUMA robot, the trade-off between storage requirements, positional resolution and positional error is discussed and an optimal system parameter scheme is derived.

1. Introduction

In the standard control technique of robot manipulators the control of the joints is done in joint coordinates, leaving it to an compiler or interpreter of the list of positioning commands to do the conversion of the external Cartesian coordinates into joint coordinates (*inverse kinematics*) in advance and to produce the list of joint coordinates. This approach hinders the developement of flexible, mobile robots.

This paper shows the approach of *learning* the inverse kinematics by using *optimal* topology-conserving mappings and discusses their resource requirements for tolerable positioning errors in the case of a PUMA robot manipulator, shown in figure 1 with a cubic workspace.

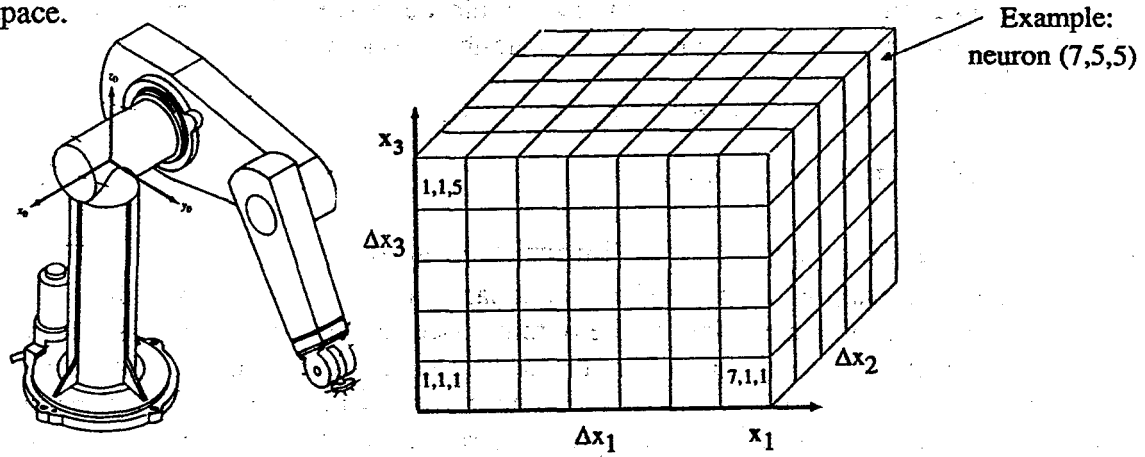


Fig. 1 The PUMA robot manipulator [FU] and a cubic workspace

2. Robot control by topology conserving mappings

One of the best known algorithms showing topology-conserving properties is the one introduced by Kohonen 1982 [KOH1] or [KOH2] and analyzed for instance by Ritter and Schulten [RITT1]. Let us now briefly describe this algorithm.

Consider as input space $X \subset \Re^3$ the Cartesian space with the input events $\mathbf{x} = (x_1,x_2,x_3)$, and an output space $\{\mathbf{y} = (i,j,k)/\ i,j,k \text{ from } 1..n\}$. So the input space is projected on an output space of descrete points y (*neurons*), determined by 3 natural numbers (indices). To each y of the output space there corresponds a set $\{\mathbf{x}\}$ of points (*a class*) of the input space. Since it is finite and bounded, the whole set of points $\{\mathbf{y}\}$ can also be ordered by one index k =1..N.

Let every point $\mathbf{y}$ (neuron) weight the input by one weight per input component, i.e. by a weight vector or *class prototype* $\mathbf{w} = (w_1,w_2,w_3)$ from X. Then the mapping of the sensor space (perhaps deformed by sensor characteristics) to the Cartesian space is done by $\mathbf{x} \mapsto \mathbf{y}_c=(i,j,k)$ with

$$|\mathbf{x}-\mathbf{w}_c| = \min_k |\mathbf{x}-\mathbf{w}_k| \tag{2.1}$$

This input-output mapping defines a neighbourhood of points $\mathbf{x}$ around every $\mathbf{w}_c$ to be mapped to the neuron $\mathbf{y}_c$. The following stochastic learning step for the weights has topology-conserving capabilities (see [KOH3]):

In the (t+1)-th iteration step, change the weight vector $\mathbf{w}_k$
for all neurons $\mathbf{y}_k$ which are in the neighbourhood of $\mathbf{y}_c$
to
$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \gamma(t+1)\, h(t+1,c,k)\, [\mathbf{x}(t+1) - \mathbf{w}_k(t)] \tag{2.2}$$

This is accomplished by the *neighbourhood function*
$$h(t, c, k) = \begin{cases} 1 & \text{if } \mathbf{y}_k \text{ is in the neighbourhood } N_c(t) \text{ of } \mathbf{y}_c \\ 0 & \text{else} \end{cases}$$

and the conditions for the *learning rate* $\gamma(t)$
$$\lim_{t\to\infty} \gamma(t) = 0, \quad \sum_{t=1}^{\infty} \gamma(t) > \infty, \quad \sum_{t=1}^{\infty} \gamma(t)^2 < \infty \tag{2.3}$$

The neighbourhood function h(.) can be varied; for instance Ritter and Schulten [RITT2] assumed h(.) to be a Gaußian-shaped function, e.g. $h(t,c,k) := \exp(-(\mathbf{y}_c-\mathbf{y}_k)^2 / 2\sigma(t)^2)$, instead of a step function. In both cases, the neighbourhood is made smaller with increasing t by decreasing the step-width or the standard deviation σ of the Gaußian distribution.
To each Cartesian position $\mathbf{y}_c=(i,j,k)$ there corresponds by a non-linear mapping a joint coordinate position $\Theta_c=(\theta_1,\theta_2,\theta_3)$ which also should be learned. Denoting $\mathbf{u}:=\Theta_c$ we get the stochastic approximation learning rule in the neighbouthood h(.) by

$$\mathbf{u}_c(t+1) = \mathbf{u}_c(t) + h(.)\gamma(t+1)[\mathbf{u}_c^*(t+1)-\mathbf{u}_c(t)] \tag{2.4}$$

with the $(t+1)^{th}$ estimation $\mathbf{u}_c^*$ of $\mathbf{u}_c$.

3. Optimal mappings and maximal information gain

Let us consider a mapping as it is defined in equation (2.1). Since sets of points of the input space are mapped to single points in the output space, there is certainly less information in the input than in the output. One plausible principle of a good mapping is to transmit as much information from the input to the output as possible (*maximal information gain principle*). This optimality criterion was proposed by Linsker [LIN1], who suggested that this might be a fundamental principle for the organization of biological neural systems.
Knowing the input pattern $\mathbf{x}$, the Shannon information gain from the N output points $\mathbf{w}_i$ is

$$I_{trans} = I_{out} - I_{out/inp} = -\ln[P(\mathbf{w}_i)] + \ln[P(\mathbf{w}_i/\mathbf{x})]$$

The average transmitted information for all inputs and outputs is with the expectation operation $\langle f(\mathbf{w}_i)\rangle := \Sigma_{\mathbf{w}_i} P(\mathbf{w}_i)\, f(\mathbf{w}_i)$

$$\langle I_{trans}\rangle_{\mathbf{w}_i,\mathbf{x}} = \langle I_{out}\rangle_{\mathbf{w}_i,\mathbf{x}} - \langle I_{out/inp}\rangle_{\mathbf{w}_i,\mathbf{x}} = -\Sigma_i P(\mathbf{w}_i)\ln[P(\mathbf{w}_i)] - \Sigma_{\mathbf{x}} P(\mathbf{x}) \Sigma_i P(\mathbf{w}_i/\mathbf{x})\ln[P(\mathbf{w}_i/\mathbf{x})]$$

The average transmitted information $\langle I_{trans}\rangle$ is maximized when

$$\langle I_{out} \rangle_{w_i,x} \overset{!}{=} \max \quad (3.1) \quad \text{and} \quad \langle I_{out/inp} \rangle_{w_i,x} \overset{!}{=} \min$$

It is easy to see [BRA] by variation analysis that (3.1) is satisfied when $P(\mathbf{w}_i) = P(\mathbf{w}_j) = 1/N$ for all i and j. Furthermore, if every input pattern $\mathbf{x}$ is only assigned to one appropriate class $\mathbf{y}_i$, we have $\langle I_{out/inp} \rangle = 0$. This means, that also for the maximal average information transmission the condition $P(\mathbf{w}_j) = 1/N$ is sufficient.

What does this mean for the *density of the classes* (number of classes per input space area unit, also called *magnification factor* $M(\mathbf{x})$) in the input space ?
It can be shown [BRA], that the condition above implies $M(\mathbf{x}) \sim p(\mathbf{x})$. In other words, for the topology conserving mapping which preserves the maximum of information *the point density of the class prototypes must approximate the probability distribution of the input patterns.*
It should be noted that this is contrary to the findings of Linsker himself in [LIN2], who stated that in optimal topology-conserving maps the often referenced classes should become bigger in the space, not smaller.
For the algorithm of section 2, Ritter and Schulten [RITT1] found that $M(\mathbf{x}) \sim p(\mathbf{x})$ is not generally true in the n-dim case. For the linear, 1-dim case they found $M(\mathbf{x}) \sim p(\mathbf{x})^{2/3}$, contrary to Kohonen [KOH2]. For the 2-dim (complex) case they also found $M(\mathbf{x}) \sim p(\mathbf{x})$. Therefore, at least for the 2-dim case, Kohonens mapping can be termed *optimal.*

For robot control the optimality criterion above is quite instructive to interprete. If we have regions of the action space where the action occur very often, this region should be better controlled and should have therefore a better resolution to minimize the average control error.

4. Position error and optimal system parameters

The positioning algorithm presented in section 2 is far too rough. Since we map a real-valued position $\mathbf{x}$ to an indexed position $\mathbf{y}_c$=(i,j,k) with a certain Θ_c, we get a positional error: For a cubic workspace with the edgelength of 70 cm and N=1000 neurons we have an error of $7x3^{1/2}$=12.12 cm which is much too high for normal robot operation. To reduce this resolution error, we approximate the true position $\Theta_{true}(\mathbf{x})$ by the sum of the coarse resolution value Θ_c and a linear approximation $\Delta\Theta = \mathbf{A}\,(\mathbf{x}-\mathbf{w})$, the first term of a Taylor expansion:

$$\Theta(\mathbf{x}) = \Theta_c + \Delta\Theta = \Theta_c + \mathbf{A}_c\,(\mathbf{x}-\mathbf{w}_c) \qquad (4.1)$$

Certainly, the matrix $\mathbf{A}_c$ is a good approximation only for a small section of the output space and is therefore different for different positions (i,j,k). With the redefinition $\mathbf{u}_c := (\theta_1,\theta_2,\theta_3,A_{11}, \ldots ,A_{33})_c^T$ we can learn both Θ_c and $\mathbf{A}_c$.
The new estimations of Θ_c and of $\mathbf{A}_c$ are obtained by using the measured error $(\mathbf{x}-\mathbf{x}_F)$ of the final position $\mathbf{x}_F$ in the linear approximation

$$\Theta_c^* = \Theta_c + \mathbf{A}_c(\mathbf{x}-\mathbf{x}_F) \qquad (4.2)$$

and $$(\mathbf{A}_c^*)_{ij} := [\theta_i(\mathbf{x}_F+dx) - \theta_i(\mathbf{x}_F)]/dx_j = [\mathbf{A}\,dx]_i / dx_j \approx [\mathbf{A}(\mathbf{x}-\mathbf{x}_F)]_i / (\mathbf{x}-\mathbf{x}_F)_j$$

which uses the fact that $\mathbf{A}$ is the first derivation in the first term of the Taylor expansion.

Nevertheless, on principle there rests a positioning error due to the linear approximation for a non-linear funktion. Let us compute this error for a linear path in the cubic workspace of a PUMA robot (see fig. 1). Let us assume that the position events are equally distributed in the workspace, the algorithm with the estimation for the joint coordinates of $\mathbf{w}_c$ has con-

verged to the true value. Then $\Theta_{c,t}$ is the true inverse kinematic transformation at $\mathbf{w}_c$; the matrix $\mathbf{A}_c$ has converged, too, and is identical to the first derivate of $\Theta_{true}(\mathbf{x})$ at $\mathbf{w}_c$. Knowing the analytical solution for the PUMA robot [FU] we can compute the maximal positioning error of the approximation for each neuron $\mathbf{y}_c$ in the linear path [BRA].
The overall maximal positioning error is determined by the superposition of two independant sources of error: the error of the linear approximation and the finite resolution n of the neuronal grid and additionally the error of the numerical resolutions r_w, r_θ and r_A (bits per stored number):

$$e^{MAX} = | e^{LA} + e^{RES} | \tag{4.1}$$

For a certain storage increment Δs the error will change by

$$\Delta e^{MAX}(s) = \frac{d}{ds} e^{MAX}(s) \; \Delta s \tag{4.2}$$

$$= [\frac{\partial}{\partial n} e^{MAX}(n) \frac{\partial}{\partial s} n(s) + \frac{\partial}{\partial r_w} e^{MAX}(r_w) \frac{\partial}{\partial s} r_w(s) + \frac{\partial}{\partial r_\theta} e^{MAX}(r_\theta) \frac{\partial}{\partial s} r_\theta(s) + \frac{\partial}{\partial r_A} e^{MAX}(r_A) \frac{\partial}{\partial s} r_A(s)] \Delta s$$

If all terms of the sum are equal, no storage rearrangement can diminish the error any more. The storage configuration can therefore be termed *optimal*. This leads us to a system of three equations with the four variables n, r_w, r_θ and r_A. In [BRA] this is solved, getting three variables as a function of the forth. By additionally using the storage equation $s=n^3 3(r_w+r_\theta+3r_A)$ we finally can calculate the maximal positioning error $e^{MAX}(s_{opt})$ as a function of the storage requirement s_{opt} by the optimal system parameters.
The optimal system parameters yield for the PUMA robot and an Cartesian positioning error of 0.201 mm, a value which is in the range of normal mechanical inaccuracy, s_{opt}=1.9 MB of storage memory contained in $N=39.6^3$ neurons with a resolution of $r=r_w=r_\theta=r_A=16.4$ Bits.

5. Conclusion

For the inverse kinematic problem in robot control the approach of learning the transformation table values has the advantages of a fast, non-analytical solution with modest memory requirements which can be used for manipulators even with many (>3) or worn-out joints and adapts to a user-defined manipulator geometry. Nevertheless, there are some associated problems: This low level approach is completely isolated in respect to higher level functions, the mapping must be relearned like an associative memory each time the workspace changes and there is no "abstract", position-independant coding of a movement.

References

[BRA] R.Brause, Performance and storage requirements of topology-conserving maps for robot manipulator control, Fachbereich Informatik report 5/89, University of Frankfurt, West Germany 1989

[FU] K.S.Fu, R.C. Gonzales, C.S.G. Lee, Robotics, McGraw Hill, 1987

[KOH1] T. Kohonen, Self-organized Formation of Topologically Correct Feature Maps, Biological Cybernetics, 1982, Vol 43, pp 59-69

[KOH2] T. Kohonen, Clustering, Taxonomy and topological Maps of Patterns, IEEE Proc. 6th Int. Conf. Pattern Recognition, Oct. 1982, pp.114-128

[KOH3] T. Kohonen, Self Organization and Associative Memory, Springer Verlag 1984

[LIN1] R.Linsker, Self-Organization in a Perceptual Network, IEEE Computer, March 1988, pp.105-117

[LIN2] R.Linsker, Towards an Organizing Principle for a layered Perceptual Network, in D. Anderson (ed), Neural Information Processing Systems, Amer. Inst. of Physics (NY), 1988

[RITT1] H.Ritter, K.Schulten, On the Stationary State of Kohonen's Self-Organizing Sensory Mapping, Biological Cybernetics, 1986, Vol 54, pp.99-106

[RITT2] H.Ritter, T.Martinetz, K.Schulten, Topology--Conserving Maps for Learning Visuo-Motor-Coordination, Neural Networks, Vol 2/3 , pp. 159-167, Pergamon Press 1989, New York