



文章编号: 1003-8728(2003)02-0226-03

神经网络在机器人运动控制中的应用研究

刘成良, 张 凯, 付 庄, 曹其新, 殷跃红

(上海交通大学 机器人研究所, 上海 200030)

刘成良

摘 要: 运动控制是人工神经网络应用于机器人控制的重要内容。本文就人工神经网络用于机器人运动学正解问题进行研究, 通过建立机器人运动学神经网络模型, 给出了相应的 BP 算法, 并对 2R、3R 和 6R 机器人运动学正解进行了系统的计算机仿真, 并结合实际任务, 在 6R 焊接机器人上进行了实验验证。

关 键 词: 人工神经网络; 机器人; 运动学控制; 计算机仿真

中图分类号: TP242 文献标识码: A

On the Application of Artificial Neural Network to Kinematic Control of a Robot

L IU Cheng-liang, ZHANG Kai, FU Zhuang, CAO Qi-xin, YN Yue-hong

(Institute of Robotics, Shanghai Jiao tong University, Shanghai 200030)

Abstract: The application of artificial neural network to kinematics control of a robot is studied. The BP algorithm is derived for the solution of forward kinematics. Computer simulation is carried out for 2R, 3R and 6R robot and satisfactory results are obtained when this algorithm was used in a 6R welding robot system.

Key words: ANN (artificial neural network); Robot; Kinematics control; Computer simulation

人工神经网络 ANN (Artificial Neural Network) 是由人工建立的以有向图为拓扑结构的动态系统, 它通过对连续或断续的输入作状态响应而进行信息处理。ANN 是有大量简单的神经元相互连接而形成的自适应非线性动态系统。每个神经元的结构和功能比较简单, 但大量神经元组合而成系统的行为却非常复杂。ANN 广泛应用于机器人的任务规划、路径规划和运动控制中。上述三个问题原则上均可采用 NN 加以实现, 不少学者在这方面进行了研究工作, 且主要集中于后两个方面^[1~4]。Chen^[5]在研究运动学 NN 算法的基础上, 给出了从关节空间到 Cartesian 空间的 J 矩阵的互逆映射关系。Kim^[6]采用 NN 算法获得了冗余度机器人的精确运动学逆解。Declercq^[7]解决了实时控制中的运动学计算问题, 给出了逆运动学独立的映射关系。Guez^[8]等用多层感知器分别对二自由度、三自由度机器人的逆运动学进行了求解, Macukow^[9]使用了改进的误差反播法。Esley^[10], guo^[11]分别用五层 PDP 和 Hopfield 网络研究了机器人和逆运动学问题。

本文结合实际任务, 就 ANN 在机器人运动学正解进行研究, 给出用神经网络解决机器人运动学正解的通用方法, 对 2-DOF、3DOF、6DOF 机器人进行了计算机仿真, 最后在 6R 焊接机器人上进行了实验验证。

1 模型建立

机器人的运动学控制主要是基于正、逆运动学计算, 不仅计算复杂且需要经常校准才能保证精度, 尤其是冗余度机器人, 由于其逆运动学求解的复杂性, 使机器人运动学控制更为困难, 因此拟这里采用 NN 实现。

1.1 单层感知器 BP 网络

运动学 NN 算法实质是求解权系数 w_{ji} , 如图 1 所示, $X = [x_1 x_2 \dots x_n]^T$ 是输入向量, $Y = [y_1 y_2 \dots y_n]^T$ 是输出向量, w_{ji} 是 x_i 到 y_j 的连接权。



图 1 BP 网络

$$S_j = \sum_{i=1}^n w_{ji} x_i - \theta_j = \sum_{i=0}^n w_{ji} x_i, Y_j = f(S_j)$$

1.2 多层感知器网络用于运动学求解

多层前馈网中, 第 1 层为输入层, 第 Q 层为输出层, 中间各层为隐层。设第 q 层 ($q = 1, 2, \dots, Q$) 的神经元个数为 n_q , 输入到第 q 层的第 i 个神经元的连接权系数为 $w_{ij}^{(q)}$, 则该网络的输入输出关系为

$$S_i^{(q)} = \sum_{j=0}^{n_q-1} w_{ij}^{(q)} x_j^{(q-1)} \quad (x_0^{(q-1)} = \theta^{(q)}, w_{i0}^{(q)} = -1)$$

$$x_i^{(q)} = f(S_i^{(q)}) = \frac{1}{1 + e^{-\mu_i^{(q)}}}$$

$$i = 1, 2, \dots, n_q \quad j = 1, 2, \dots, n_{q-1} \quad q = 1, 2, \dots, Q$$

$$Y_j = f(S_j)$$

收稿日期: 2001-05-31

作者简介: 刘成良(1964-), 男(汉), 山东, 教授, 博士

E-mail: zhangkai@sytu.edu.cn



设给定 P 组输入—输出样本:

$$\Phi_p^{(0)} = [\Phi_{p1}^{(0)}, \Phi_{p2}^{(0)}, \dots, \Phi_{p,n}^{(0)}]^T \quad (p = 1, 2, \dots, P)$$

$$T_p = [T_{p1}, T_{p1}, \dots, T_{p,nq}]^T$$

连接权系数 w_{ij} 的学习方法: 用给定的 P 组输入输出, 对 BP 进行训练即对 w_{ij} 进行学习调整, 最终确定实现给定的输入输出映射关系。经过训练的 BP 网对不是样本集中的输入也能给出足够精度的输出, 这正是 NN 的泛化 (generalization) 功能。

设拟合误差代价函数为

$$E = \frac{1}{2} \sum_{p=1}^P \sum_{i=1}^{n_q} (T_{pi} - \Phi_{pi}^{(Q)})^2 = \sum_{p=1}^P E_p$$

即
$$E_p = \frac{1}{2} \sum_{i=1}^{n_q} (T_{pi} - \Phi_{pi}^{(Q)})^2$$

为求解连接权系数 w_{ij} 以使代价函数 E 最小, 这里采用一阶梯度法, 即最速下降法。一阶梯度法寻优的关键是计算优化目标函数 E 对 w_{ij} 一阶导数, 即

$$\frac{\partial E}{\partial w_{ij}^{(q)}} \quad (q = Q, Q - 1, \dots, 1)$$

$$\frac{\partial E}{\partial w_{ij}^{(q)}} = \sum_{p=1}^P \frac{\partial E_p}{\partial w_{ij}^{(q)}}$$

对 Q 层有

$$\frac{\partial E}{\partial w_{ij}^{(Q)}} = \frac{\partial E_p}{\partial \Phi_{pi}^{(Q)}} \frac{\partial \Phi_{pi}^{(Q)}}{\partial s_{pi}^{(Q)}} \frac{\partial s_{pi}^{(Q)}}{\partial w_{ij}^{(Q)}} = - (T_{pi} - \Phi_{pi}^{(Q)}) f'(s_{pi}^{(Q)}) \Phi_{pj}^{(Q-1)} = - \delta_{pi}^{(Q)} \Phi_{pj}^{(Q-1)}$$

其中

$$\delta_{pi}^{(Q)} = - \frac{\partial E_p}{\partial \Phi_{pi}^{(Q)}} = (T_{pi} - \Phi_{pi}^{(Q)}) f'(s_{pi}^{(Q)})$$

对 $Q - 1$ 层有

$$\frac{\partial E}{\partial w_{ij}^{(Q-1)}} = \frac{\partial E_p}{\partial \Phi_{pi}^{(Q-1)}} \frac{\partial \Phi_{pi}^{(Q-1)}}{\partial w_{ij}^{(Q-1)}} =$$

$$\left(\sum_{k=1}^{n_q} \frac{\partial E_p}{\partial \Phi_{pk}^{(Q)}} \frac{\partial \Phi_{pk}^{(Q)}}{\partial \Phi_{pi}^{(Q-1)}} \right) \frac{\partial \Phi_{pi}^{(Q-1)}}{\partial w_{ij}^{(Q-1)}} =$$

$$\left(\sum_{k=1}^{n_q} - \delta_{pk}^{(Q)} w_{ki}^{(Q)} \right) f'(s_{pi}^{(Q-1)}) \Phi_{pj}^{(Q-2)} = - \delta_{pi}^{(Q-1)} \Phi_{pj}^{(Q-2)}$$

其中

$$\delta_{pi}^{(Q-1)} = - \frac{\partial E_p}{\partial \Phi_{pi}^{(Q-1)}} = \left(\sum_{k=1}^{n_q} \delta_{pk}^{(Q)} w_{ki}^{(Q)} \right) f'(s_{pi}^{(Q-1)})$$

依次类推得

$$\frac{\partial E}{\partial w_{ij}^{(Q)}} = - \delta_{pi}^{(Q)} \Phi_{pj}^{(Q-1)}$$

$$\frac{\partial E}{\partial w_{ij}^{(Q-1)}} = - \delta_{pi}^{(Q-1)} \Phi_{pj}^{(Q-2)}$$

$$\frac{\partial E}{\partial w_{ij}^{(1)}} = - \delta_{pi}^{(1)} \Phi_{pj}^{(0)}$$

为反向递推计算公式。

先算出 $\delta_{pi}^{(Q)}$, 递推计算出 $\delta_{pi}^{(Q-1)}$, $\delta_{pi}^{(Q-2)}$, ..., $\delta_{pi}^{(1)}$ 。

$\delta_{pi}^{(q)}$ 中存在导数项 $f'(s_{pi}^{(q)})$ 。这里设 $f(\cdot)$ 为 $\sin(\cdot)$ 函数,

则

$$\Phi_{pi}^{(q)} = f(s_{pi}^{(q)}) = \sin(s_{pi}^{(q)})$$

$$f'(s_{pi}^{(q)}) = \cos(s_{pi}^{(q)})$$

这里归纳出该 BP 网的学习算法如下:

$$w_{ij}^{(q)}(k+1) = w_{ij}^{(q)}(k) + \alpha \delta_{ij}^{(q)}(k+1), \alpha > 0$$

$$D_{ij}^{(q)} = \sum_{p=1}^P \delta_{pi}^{(q)} \Phi_{pj}^{(q-1)}$$

$$\delta_{pi}^{(q)} = \left(\sum_{k=1}^{n_{q+1}} \delta_{pk}^{(q+1)} w_{ki}^{(q+1)} \right) \cos(s_{pi}^{(q-1)})$$

$$\delta_{pi}^{(Q)} = (T_{pi} - \Phi_{pi}^{(Q)}) \cos(s_{pi}^{(Q)})$$

因此可求出权矩阵 w_{ij} 。

2 正运动学模型

机器人手部结构如图 2 所示, 其姿态表示为

$$T = [\vec{n} \ \vec{o} \ \vec{a} \ \vec{p}] = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

式中: \vec{a} 为机器人手部接近矢量; \vec{o} 为手部姿态矢量; \vec{n} 为手部法向矢量, 此三个矢量构成了右手矢量积, 即 $\vec{n} = \vec{o} \times \vec{a}$ 。手部位置可以用从基准参考系原点指向手部中心的矢量 \vec{p} 来表示, p_x, p_y, p_z 是手部在基准参考坐标系中的坐标。

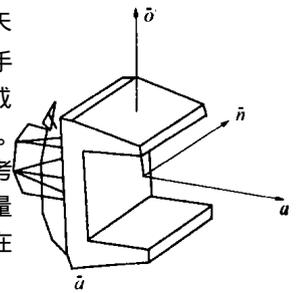


图 2 机器人手部姿态

正运动学模型即给出机器人各个关节变量 θ , 求手部姿态各矢量 $\vec{n}, \vec{o}, \vec{a}$ 和 \vec{p} 。

假设运动学正解非线性系统 $y = f(x)$, 输入输出样本空间为 $\{\Phi, T\}$, 即 $\Phi = [\theta_1(t), \theta_2(t), \dots, \theta_n(t)]^T, T = [n(t), o(t), a(t), p(t)]$ 。

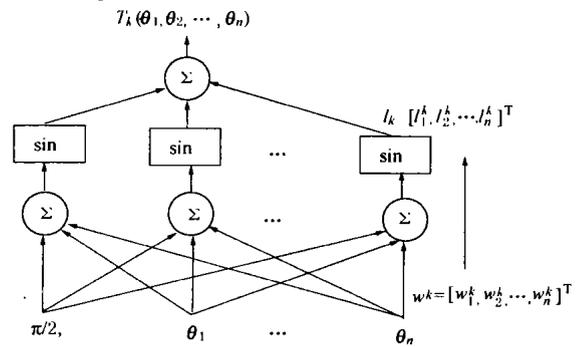


图 3 神经网络结构图

如图 3 所示, 设机器人由 n 个关节组成, 则正运动学解必为这些关节角的三角函数的组合, 可写成如下形式, 即

$$T_k(\theta) = \sum_{i=1}^m l_i^k \sin [w_i^k \theta^T], k = 1, 2, 3, \dots, l$$

式中: T_k 表示末端位姿的第 k 个分量, $\theta = [\pi/2, \theta_1, \theta_2, \dots, \theta_n]^T$ 。

当关节为移动关节时, 相应的 θ 取为常数, w_i^k 由 E 最小

的代价函数法叠代求得, l_i^k 可用如下最小方差算法计算, 即

$$l_i^k(j+1) = l_i^k(j) + \eta(T_k^d - T_k(\theta)) \sin[(w_{i,j}^k)^T \theta]$$

式中: T_k^d 为期望值; 末端位第 k 个分量; θ 为期望的关节变量; η 为学习率。

3 运动学正解BP 算法应用

3.1 $n = 2$ 的两自由度机器人

输入变量: θ_1, θ_2

输出变量: x, y

输入与输出之间关系

$$x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2)$$

$$y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

隐层节点数选择为: $m = 3^2 = 9$; 权系数: $w_{i,j} l_{i,j-1}$ ($i = 9, j = 3$)。学习样本选用 1000 组输入输出, 训练结果如图 4 和图 5 所示。由图可以看出训练结果在 300~700 次之间, 样本学习结果吻合性很强。

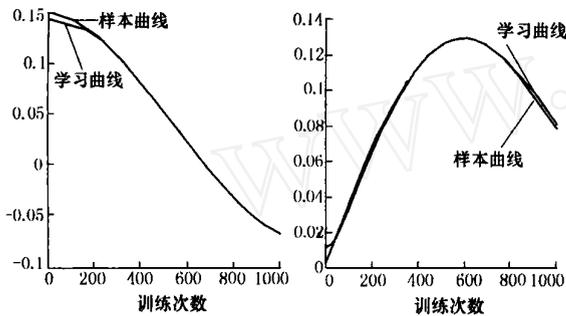


图 4 P_x 的样本及训练数据曲线 图 5 P_y 的样本及训练数据曲线

$$w_{ij} = \begin{bmatrix} 0.4591 & 0.0314 & 0.2867 \\ 0.1144 & 0.2984 & 0.7647 \\ 0.7427 & 0.1032 & 0.3592 \\ 0.5802 & 0.1118 & 0.6836 \\ 0.6824 & 0.2987 & 0.1189 \\ 0.1287 & 0.4980 & 0.3460 \\ 0.3705 & 0.6102 & 0.7201 \\ 0.2711 & 0.3346 & 0.7667 \\ 0.2331 & 0.1339 & 0.3625 \end{bmatrix}$$

$$l_{i,j-1} = \begin{bmatrix} 0.0487 & 0.3120 \\ -0.2416 & -0.2855 \\ 0.0375 & -0.3052 \\ 0.4555 & -0.3102 \\ -0.1659 & 0.1409 \\ -0.1655 & 0.0617 \\ -0.2855 & 0.1604 \\ 0.4161 & 0.2742 \\ 0.0362 & 0.5343 \end{bmatrix}$$

3.2 $n = 6$ 的六自由度机器人

输入变量 6 个: $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$

输出变量 12 个:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

物体在空间的方位有三个自由度, 但 T 中表示方位的有 9 个参数。在机器人示教中要输入 9 个量, 并且还要符合正交条件, 因此不具有实用性。这里采用侧滚角 α 俯仰角 β 偏航角 γ 定义末端执行器的三个方向角:

$$\alpha = \arctan \frac{n_x}{n_z}; \gamma = \arctan \frac{a_z}{a_x}$$

$$\beta = \arctan \frac{-n_z}{\sqrt{n_x^2 + n_y^2}}$$

选取隐层节点数: $m = 27$, 学习样本选用 1800 组输入输出。训练结果如图 6 和图 7 所示, 图中细实线代表样本数据, 粗实线代表学习数据。由图可以看出训练结果在 400~900 次之间, 样本学习结果吻合性很强。

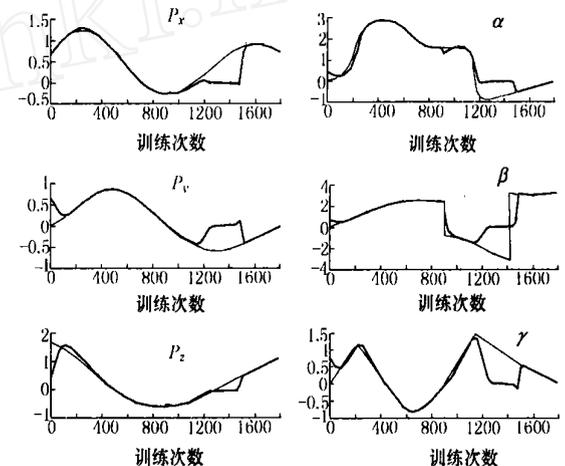


图 6 P_x, P_y, P_z 样本及训练曲线 图 7 α, β, γ 样本及训练曲线

4 实验验证

利用上述理论, 对 6R 焊接机器人进行控制实验, 实现了机器人的平稳移动及精确焊接。图 8 为机器人焊接 MECO 汽车横梁两端外环缝的焊接及计算机仿真。

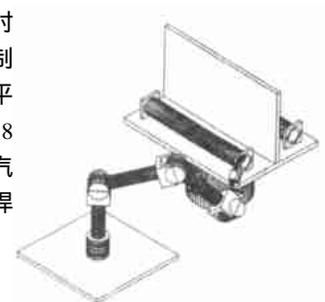


图 8 横梁焊接工作站

5 结束语

本文结合实际任务, 就 ANN 在机器人运动学

正解进行研究, 给出用神经网络解决机器人运动学正解的通用方法, 对 2-DOF、3-DOF、6DOF 机器人进行了计算机仿真, 由仿真结果可以看出, 采用上述算法学习正运动学可以获得精确的正运动学解, 最后在 6R 焊接机器人上进行了实验验证。

(下转第 331 页)

表 1 机床床头箱物理模型的部分信息

参数符号	参数类型	参数名称	参数值或导出公式	单位	箱体	电机	轴系 I	轴系 II	轴系 III	主轴
n_{motor}	继承	电机额定转速	1440	r/min		Δ				
n_{min}	继承	主轴最低转速	31.5	r/min		Δ	Δ	Δ	Δ	Δ
R_n	继承	主轴转速范围	43.79			Δ	Δ	Δ	Δ	Δ
n_{css1}	继承	轴 I 计算转速	1000	r/min			Δ			
P_1	导出	轴 I 传递的功率	$P_1 = P \times \eta_{belt}$				Δ			
T_1	导出	轴 I 传递的转矩	$T_1 = 9549 \frac{P_1}{n_{css1}}$				Δ			
d_{1r}	导出	轴 I 按扭转刚度条件的估算轴径	$d_{1r} = 91 \times \sqrt[3]{\frac{P_1}{n_{css1} \times [Q]}}$				Δ			
$m_{wz1,z2}$	导出	齿轮副 z_1, z_2 按弯曲疲劳强度估算的模数	$m_{wz1,z2} = 32 \times \sqrt[3]{\frac{P_{z1}}{z_1 \times n_{jz2}}}$				Δ			
$m_{jz1,z2}$	导出	齿轮副 z_1, z_2 按齿面点蚀疲劳强度估算的模数	$m_{jz1,z2} = \frac{740}{z_1 + z_2} \times \sqrt[3]{\frac{P_{z1}}{n_{jz2}}}$				Δ			

4 结束语

基于装配信息的产品物理模型很好地解决和处理了产品信息模型中的物理概念和物理信息的信息结构, 建模方法和应用方式, 成为 CAD 系统及其产品设计自动化领域新的技术发展的方向和设计自动化的新技术。本文中的产品物理模型在产品中的成功应用实例, 说明这一模型所带来的设计效率的提高, 会极大地解放设计人员的脑力和体力劳动, 使其集中精力解决关键的技术问题, 提高产品的设计技术含量, 创造出美妙的新的产品世界。

[参考文献]

[1] 张申生, 周雄辉 特征的生成转换——实现信息共享与并行工程[A]. 全国第七届 CAD 与图形学学术会议论文集[C]

[2] 张申生, 侯晓林 基于知识的开放型产品信息建模方法研究 [A]. 第三届中国计算机集成制造系统 (CMS) 学术会议 CMS-CHINA '94 论文集[C], 1994

[3] 侯晓林, 张申生 面向并行工程的产品装配模型和虚拟制造技术的研究[J]. 机械科学与技术, 1995, (5): 122~ 126

[4] Hou X L., Zhang S S. A assembly model for concurrent engineering [A]. **Proceedings of CAD/Graphics'95** [C], Wuhan, 1995, 10

[5] Zhang S S, Hou X L. STEP application protocol interoperability for CAD/CAPP integration [A]. **Proceedings of ASME Computer in Engineering**, Chicago U. S. A. 1995, (10)

[6] Zhang S S, Hou X L. Product modeling technology for computer aided concurrent design [A]. **Proceedings of CAPE'95** [C]

(上接第 228 页)

[参考文献]

[1] 孙增圻编著 智能控制理论与技术[M]. 北京: 清华大学出版社, 1997

[2] 金耀初, 蒋静坪 神经网络在机器人控制中的应用[J]. 机器人, 1999, 14(6)

[3] 张伟 基于神经网络的机器人位姿逆解[J]. 机器人, 1997, 19(2)

[4] 王正志, 周宗谭 MRS 自组织三维神经网络的改进及其在机器人控制中的应用[J]. 自动化学报, 1992, 18(1): 99~ 104

[5] Chen C L., Philip, McAulay A D. Polynomial neural network for robot forward and inverse kinematics learning computations[A]. **Proceedings of SPIE-The International Society for Optical Engineering Applications of Artificial Intelligence**[C], IX Apr 2- 4 1991: 394~ 405

[6] Kim Ki-Kap, Yoon Yong-San Practical inverse kinematics of a kinematically redundant robot using a neural network [J]. **Source: Advanced Robotics**, 1992, 6(4): 431~ 440

[7] Declercq F, Dumortier F, De Keyser R, Van Cauwenbergh A. Real-time control of a robot using neural networks[A]. **Proceedings of the IEEE Conference on Control Applications**[C]. Part 2 (of 3), Aug 24- 26, 1994, (2)

[8] Guez A. et al. Solution to the inverse kinematics problem in robotics by neural networks[J]. **IEEE Conf. on Neural Networks**, 1988, (2): 584~ 587

[9] Macukow M. Robot control with neural networks[A]. **AI and Information Control Systems of Robots**[C], Plander North-Holland, 1989

[10] Esley R. A learning architecture for control based on BP neural networks[A]. **Proc IJCNN**[C], 1988, 2: 587~ 594

[11] Guo J. A solution to the inverse kinematics in robotics using neural networks[A]. **Proc IJCNN**[C], 1989, 1