

[研究·设计]

DOI:10.3969/j.issn.1005-2895.2015.02.004

# 基于非线性偏最小二乘的操作臂逆解算法

费冬冬<sup>1</sup>, 邓琛<sup>1</sup>, 张琴舜<sup>2</sup>, 黄祖良<sup>1</sup>

(1. 上海工程技术大学 电子电气工程学院, 上海 201620;  
2. 上海交通大学 核科学与工程学院, 上海 200240)

**摘要:**操作臂逆运动学问题是机器人控制中的一项重要内容。目前使用较多的神经网络法大多为多输入多输出或者多输入单输出方式,需要大量运算。非线性偏最小二乘法(NLPLS)建立的模型分为内部和外部模型,样本数据经外部模型处理后才用于训练若干个单输入单输出的神经网络。对PUMA560操作臂的仿真试验表明,在相同隐层神经元数的情况下,该算法比普通神经网络法具有更好的预测精度。这也表明,NLPLS只需较少的隐层神经元数就可以达到普通方法的精度,从而减少运算量。

**关键词:**机器人;操作臂;非线性偏最小二乘;逆运动学

**中图分类号:**TP241.2 **文献标志码:**A **文章编号:**1005-2895(2015)02-0014-05

## Inverse Kinematics Solution of Manipulator Based on Nonlinear PLS

FEI Dongdong<sup>1</sup>, DEN Chen<sup>1</sup>, ZHANG Qinshun<sup>2</sup>, HUANG Zuliang<sup>1</sup>

(1. School of Electronic and Electric Engineering, Shanghai University of Engineering Science, Shanghai 201620, China;  
2. School of Nuclear Science and Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

**Abstract:** The inverse kinematics of manipulator is a vital problem in robot control. Generally, most of the neural networks are applied by many researchers as numerical solutions with multi-input-multi-output or multi-input-single-output ones, and that means a large amount of computations. Models built by NLPLS were divided into outer models and inner models. The samples were used for the training process of several neural networks after they were processed by outer models. Simulations based on the PUMA 560 manipulator show that NLPLS has a better prediction accuracy than ordinary solutions when they have the same number of hidden units. The results also indicate that, by applying NLPLS, less computation is needed to meet the same accuracy requirements.

**Key words:** robot; manipulators; nonlinear partial squares (NLPLS); inverse kinematics

机器人逆运动学问题是近年来的研究热点,求运动学逆解得到机械臂关节从笛卡尔空间到关节空间的映射,其方法可分为封闭解法和数值解法两类。封闭解法一般适用于6 DOF下的操作臂,且推导过程一般比较复杂<sup>[1-2]</sup>。数值解法是一种基于迭代的解法,相对封闭解法,它更具有一般性。其中,神经网络由于能够以任意精度拟合任意非线性函数<sup>[3]</sup>,而得到了研究人员的青睐,很多研究与之有关<sup>[4-8]</sup>。王海鸣<sup>[9]</sup>等人利用Levenberg-Marquardt算法避免了训练中可能陷入局部最小值的情况。Rasit Köker<sup>[10]</sup>将神经网络与遗传算

法结合,用训练好的多个网络对同一输入的输出进行预测,经遗传算法对各预测值进行处理,提高了模型的预测精度。Yunduan Cui<sup>[11]</sup>等人提出了四元数神经网络法,相比于实数神经网络,该方法在混沌时间序列的预测问题上表现出色。但上述神经网络都属于多输入多输出模型,包含较复杂的运算,对处理速度要求较高。而非线性偏最小二乘法与普通神经网络模型的不同在于,位置坐标矩阵和关节角矩阵首先经过线性偏最小二乘法处理得到主成分向量,然后再用于训练神经网络。由于这些神经网络都是单输入单输出的,因

收稿日期:2014-09-12;修回日期:2014-11-11

基金项目:国家自然科学基金项目(61201244);中央财政专项资金项目资助(ye-13-3-2);上海工程技术大学研究生创新项目资助(E1-0903-14-01036)

作者简介:费冬冬(1990),男,江苏南通人,硕士研究生,主要研究方向为嵌入式系统、机器人等。E-mail:fei-dongdong@163.com

此运算量相对较少,且模型还具有偏最小二乘法的鲁棒能力,仿真试验结果表明它比普通神经网络方法具有更高的预测精度。

## 1 非线性偏最小二乘法

非线性偏最小二乘法所建立的模型分外部模型和内部模型,线性偏最小二乘回归模型为外部模型,描述其输出的  $t$  (自变量主成分) 和  $u$  (因变量主成分) 之间的非线性函数由内部模型(神经网络)进行拟合。内部模型描述的非线性回归模型可表示为

$$u_i = N(t_i) + r_i \quad (1)$$

式中:  $N(\cdot)$  表示第  $i$  对主成分向量  $u_i$  和  $t_i$  的非线性关系,  $r_i$  表示残差。由于内部模型采用的是神经网络,所以这种形式的 NNPLS 也被称为神经网络偏最小二乘法(NNPLS)。

其算法原理图如图 1 所示,令  $E_0 = X, F_0 = Y$ , 输入至 PLS 外部模型(1), 提取第 1 对主成分  $u_1$  与  $t_1$ , 接下来以  $u_1$  为输出,  $t_1$  为输入训练神经网络(1), 建立第 1 对主成分之间的非线性关系, 然后计算残差矩阵  $E_1$  与  $F_1$ , 作为 PLS 外部模型(2)的输入。重复这些步骤, 直至提取完所有的主成分。从中可以看出, NNPLS 算法需要建立多个神经网络模型。其数量等于提取的主成分数, 而主成分数不超过矩阵  $X$  的秩。运用于操作臂中,  $X$  矩阵的每一行代表一个样本, 每一行的各个元素分别代表决定机械臂位姿的 12 个参数;  $Y$  矩阵的每一行与  $X$  矩阵的同一行对应, 每一行的各个元素分别为机械臂各个关节的转动角度。机械臂末端执行器的位姿由 12 个参数决定, 所以位姿矩阵  $X$  的秩不会超过 12。如果只考虑末端执行器的位置, 不考虑姿态, 那么这个数字将减少到 3, 至多建立 3 个神经网络模型即可。

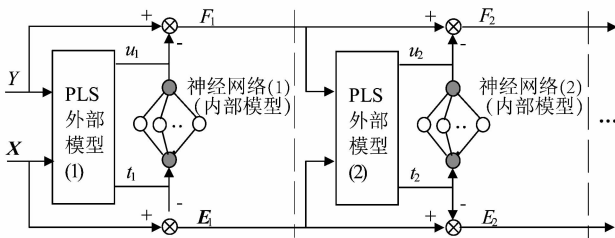


图 1 NNPLS 算法原理图

Figure 1 Schematic diagram of NNPLS

NNPLS 算法将多元回归问题分解为若干个一元回归问题, 每次只有一个单输入单输出的神经网络参与训练, 因此在相同硬件条件下能够缩短处理时间。

1) NNPLS 的具体算法<sup>[12-13]</sup> ( $i$  初始值为 1):

① 令  $E_{i-1} = X, F_{i-1} = Y$ , 分别求  $E_{i-1}^T F_{i-1}, F_{i-1}^T E_{i-1}$  和  $F_{i-1}^T E_{i-1} E_{i-1}^T F_{i-1}$  的最大特征值  $\lambda_i$  对应的特征向量  $w_i$  及  $c_i$ ;

② 计算主成分向量  $t_i$  和  $u_i$

$$\begin{cases} t_i = E_{i-1} w_i / (w_i^T w_i) \\ u_i = F_{i-1} c_i / (c_i^T c_i) \end{cases}; \quad (2)$$

③ 计算负荷向量  $p_i$  和  $q_i$

$$\begin{cases} p_i = E_{i-1}^T t_i / (t_i^T t_i) \\ q_i = F_{i-1}^T u_i / (u_i^T u_i) \end{cases}; \quad (3)$$

④ 将  $t_i$  作为神经网络  $i$  的输入,  $u_i$  作为输出, 对神经网络  $i$  进行训练, 得到第  $i$  个神经网络模型  $N_i(\cdot)$ ;

⑤ 令  $u_i = N_i(t_i)$ , 计算残差

$$\begin{cases} E_i = E_{i-1} - t_i p_i^T \\ F_i = F_{i-1} - u_i q_i^T \end{cases}; \quad (4)$$

⑥ 判断  $i$  是否大于  $a$ , 若成立则结束, 否则令  $i = i + 1, X = E_i, Y = F_i$ , 返回第①步。其中  $a$  为所选成分数。

2) 建立好  $a$  个神经网络模型后, 对给定的  $X$ , 可按下述方法预测  $\hat{Y}$ :

① 计算  $a$  个主成分向量  $t_i$

$$t_i = X w_i^*, \quad (5)$$

$$w_i^* = (I - w_1 p_1^T)(I - w_2 p_2^T) \cdots (I - w_{i-1} p_{i-1}^T) w_i, \quad (6)$$

式中  $I$  为单位矩阵;

② 用训练好的  $a$  个神经网络预测相应的主成分向量  $u_i$ ;

③ 最后按如下公式计算可得:

$$\hat{Y} = u_1 q_1^T + u_2 q_2^T + \cdots + u_a q_a^T. \quad (7)$$

## 2 PUMA 560 机器人

PUMA 560 是 Unimation 公司的一款工业机器人, 在工业和学术研究中有着广泛的应用。本文亦以 PUMA 560 为例, 用 NNPLS 算法求其逆解。其运动参数和坐标系的分布如图 2 所示, 其中坐标系 {2} 与坐标系 {1} 原点重合, 如图中曲线箭头所示。表 1 给出了图 2 坐标系分布下的 D-H 参数<sup>[14]</sup>。

若只考虑 PUMA 560 末端执行器的位置, 不考虑姿态, 经推导可知末端执行器的  $x, y, z$  坐标与关节角之间的关系为<sup>[15]</sup>:

$$p_x = c_1(a_2 c_2 + a_3 c_{23} - d_4 s_{23}) - d_3 s_1, \quad (8)$$

$$p_y = s_1(a_2 c_2 + a_3 c_{23} - d_4 s_{23}) + d_3 c_1, \quad (9)$$

$$p_z = -a_3 s_{23} - a_2 s_2 - d_4 c_{23}. \quad (10)$$

式中:  $c_1 = \cos \theta_1, s_1 = \sin \theta_1, c_{23} = \sin(\theta_2 + \theta_3)$ ,

$s_{23} = \sin(\theta_2 + \theta_3)$ , 以此类推;  $\theta_i$  为关节角,  $\alpha_{i-1}$  为连杆转角,  $a_{i-1}$  为连杆长度,  $d_i$  为连杆偏距。由公式(8)和公式(9)可知, PUMA 560 末端三维坐标值只与  $\theta_1, \theta_2, \theta_3$  有关。

表1 PUMA 560 的 D-H 参数

Table 1 Denavit hartenberg parameters of PUMA 560

| $i$ | 连杆转角<br>$\alpha_{i-1}/\text{rad}$ | 连杆长度<br>$a_{i-1}/\text{m}$ | 连杆偏距<br>$d_i/\text{m}$ | 关节角<br>$\theta_i/(\circ)$ |
|-----|-----------------------------------|----------------------------|------------------------|---------------------------|
| 1   | 0                                 | 0.000 0                    | 0.000 00               | -160 ~ 160                |
| 2   | $\pi/2$                           | 0.000 0                    | 0.000 00               | -225 ~ 45                 |
| 3   | 0                                 | 0.431 8                    | 0.150 05               | -45 ~ 225                 |
| 4   | $-\pi/2$                          | 0.020 3                    | 0.431 80               | -110 ~ 170                |
| 5   | $\pi/2$                           | 0.000 0                    | 0.000 00               | -100 ~ 100                |
| 6   | $-\pi/2$                          | 0.000 0                    | 0.000 00               | -266 ~ 266                |

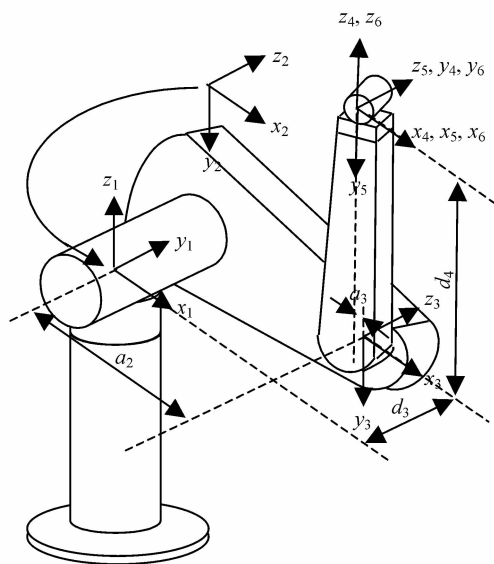


图2 PUMA 560 坐标系分布及运动参数

Figure 2 Frame assignments and kinematic parameters of PUMA 560

### 3 仿真试验

仿真试验在 MATLAB R2013a 环境下进行, 图3 为其仿真界面。构造训练样本时, 首先在  $\theta_1, \theta_2, \theta_3$  的取值区间内每间隔  $5^\circ$  取 1 个值, 即  $\theta_1$  取 65 个值,  $\theta_2$  取 55 个值,  $\theta_3$  取 55 个值, 然后将 3 组角度值排列组合, 得到 196 625 组角度值样本。每组角度值根据公式(8)、公式(9)和公式(10), 以及表1 中的 D-H 参数, 计算相应的三维坐标值。用角度样本构造  $Y$  矩阵, 其每一行代表一个样本, 每一列分别代表  $\theta_1, \theta_2, \theta_3$  值; 以三维坐标样本构造  $X$  矩阵, 每一行与  $Y$  矩阵的相同行对应, 每一列元素代表操作臂的三维坐标。

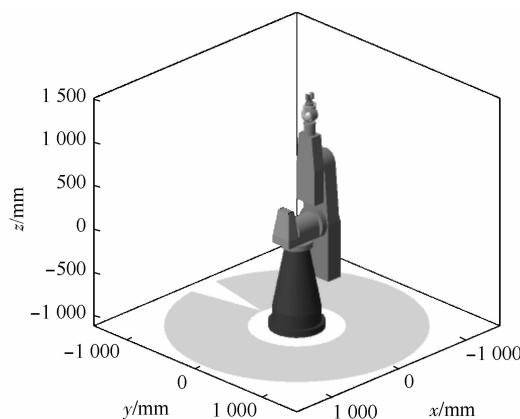


图3 PUMA 560 仿真界面

Figure 3 Simulation interface of PUMA 560

构造好 2 个矩阵后, 用 NNPLS 算法拟合  $X$  与  $Y$  之间的非线性关系,  $X$  为自变量,  $Y$  为因变量。主成分数  $a$  一般不超过  $X$  的秩, 而这里  $X$  的秩不大于 3。经过测试, 当取  $a = 3$  时, 效果最好。本文采用改进的 BP 神经网络——Levenberg-Marquardt 法, 提高网络的收敛速度, 同时避免陷入局部最小值。3 个神经网络都选用对数 S 型函数作为传递函数, 训练精度目标设为 0.003, 隐层 300 个神经元, 输出层 1 个神经元, 最大迭代次数 100 000 次。

为了与普通神经网络法对比, 相同的样本还被直接用于训练 Levenberg-Marquardt 网络, 相应神经网络设置为: 输入、输出层各 3 个神经元, 隐层 300 个神经元, 传递函数为对数 S 型函数, 训练精度目标 0.003, 最大迭代次数 100 000 次。

表 2 和表 3 为 NNPLS 算法与普通神经网络 (L-M) 方法的仿真结果对比。从表 2 和表 3 可以看出, 直接对训练样本采用神经网络的预测精度比 NNPLS 低, 在第 4 组试验中, 普通方法的相对误差绝对值的均值达到了 NNPLS 的 12 倍, 说明 NNPLS 模型比普通神经网络方法具有更好的预测精度。一般而言, 隐层神经元数越多, 模型的预测效果越好, 所以仿真结果也表明, NNPLS 只需较少的隐层神经元数就可达到普通神经网络的预测效果。

试验中也发现 NNPLS 模型的某些预测值的相对误差达到了 50% 以上, 其原因除了模型本身的误差外, 与训练的样本也有关。因为对于一个给定的笛卡尔空间坐标, 其对应的关节空间的解不止一组, 这些解之间差别可能很大。一组解和操作臂末端坐标之间代表着一种非线性关系, 若有 8 组解, 就有 8 种不同的非线性关系, 这些样本同时用于一个只能反映一种关系

的神经网络的训练,势必造成神经网络无法准确拟合输入和输出之间的映射关系;另外,PLS模型的预测精度可能会随着样本容量的增大而降低<sup>[16]</sup>。改进的方法是对输入 NNPLS 模型的样本进行预处理,对那些关

节角之间差别很大,但是对应末端之间相距很近的样本点只保留其中的某一个;把操作臂工作空间分为若干个子区域,在这些子区域上的样本分别进行建模,以此提高预测精度。

表2 L-M 与 NNPLS 算法比较

Table 2 Comparison of L-M and NNPLS algorithm

| 组号 | 算法    | $\theta_1/\text{rad}$ |          | $\theta_2/\text{rad}$ |          | $\theta_3/\text{rad}$ |          |
|----|-------|-----------------------|----------|-----------------------|----------|-----------------------|----------|
|    |       | 预测值                   | 实际值      | 预测值                   | 实际值      | 预测值                   | 实际值      |
| 1  | L-M   | -0.448 1              | -0.436 3 | -0.415 1              | -0.436 3 | -0.479 8              | -0.436 3 |
|    | NNPLS | -0.395 0              | -0.436 3 | -0.435 4              | -0.436 3 | -0.446 7              | -0.436 3 |
| 2  | L-M   | -0.476 5              | -0.436 3 | -0.114 8              | -0.087 2 | -0.471 9              | -0.436 3 |
|    | NNPLS | -0.407 5              | -0.436 3 | -0.085 0              | -0.087 2 | -0.440 9              | -0.436 3 |
| 3  | L-M   | -0.421 0              | -0.436 3 | 0.209 6               | 0.261 7  | -0.460 5              | -0.436 3 |
|    | NNPLS | -0.420 7              | -0.436 3 | 0.265 2               | 0.261 7  | -0.434 3              | -0.436 3 |
| 4  | L-M   | 0.187 2               | -0.087 2 | -0.377 9              | 0.261 7  | 0.253 8               | -0.087 2 |
|    | NNPLS | 0.262 5               | -0.087 2 | -0.438 9              | 0.261 7  | 0.254 7               | -0.087 2 |

表3 L-M 与 NNPLS 误差的比较

Table 2 Error comparison of L-M and NNPLS

| 组号 | 算法    | 相对误差/%     |            |            | M/%   |
|----|-------|------------|------------|------------|-------|
|    |       | $\theta_1$ | $\theta_2$ | $\theta_3$ |       |
| 1  | L-M   | 2.69       | -4.86      | 9.96       | 5.84  |
|    | NNPLS | -9.46      | -0.22      | 2.37       | 4.02  |
| 2  | L-M   | 9.20       | 31.52      | 8.15       | 16.29 |
|    | NNPLS | -6.60      | -2.63      | 1.05       | 3.42  |
| 3  | L-M   | -3.50      | -19.92     | 5.53       | 9.65  |
|    | NNPLS | -3.59      | 1.31       | -0.48      | 1.79  |
| 4  | L-M   | -28.48     | -13.39     | -1.35      | 14.41 |
|    | NNPLS | 0.26       | 0.58       | -2.70      | 1.18  |

说明:M 为相对误差绝对值的平均值

## 4 结语

本文提出了一种求解操作臂逆运动学解的新模型,对模型优缺点进行了分析,并提出了模型改进意见。该模型分为外部模型和内部模型,外部模型为线性偏最小二乘法,内部模型为改进的 BP 神经网络。样本由操作臂正运动学公式构造,在关节角取值范围内选取一系列值代入正运动学公式得到末端对应三维坐标。经外部模型处理后再用于训练若干个单输入单输出的神经网络,仿真试验结果表明,NNPLS 在相同隐层神经元数的情况下,比普通神经网络法具有更好的预测精度。这也表明 NNPLS 只需较少的隐层神经元数就可以达到普通神经网络的预测效果,可以在一定程度上减少计算量,从而加快逆解计算速度。

## 参考文献:

- [1] 倪振松,廖启征,魏世民,等.空间 6R 机器人位置反解的对偶四元数法[J].机械工程学报,2009,45(11):25-29.
- [2] 杭鲁滨,王彦,杨廷力.基于 Groebner 基法的一般串联 6R 机器人

机构逆运动学分析[J].上海交通大学学报,2004,38(6):853-856.

- [3] XING Jingjing, LUO Ruimin, GUO Haili, et al. Radial basis function network-based transformation for nonlinear partial least-squares as optimized by particle swarm optimization; application to QSAR studies [J]. Chemometrics and Intelligent Laboratory Systems, 2014, 130:37-44.
- [4] GUO Dongsheng, ZHANG Yunong. Zhang neural network, Getz-Marsden dynamic system, and discrete-time algorithms for time-varying matrix inversion with application to robots' kinematic control [J]. Neurocomputing, 2012, 97:22-32.
- [5] SHAH J A, RATTAN S S, NAKRA B C. Kinematic analysis of 3-DOF planar robot using artificial neural network [J]. International Journal of Robotics and Automation, 2012, 1(3):145-151.
- [6] CAI Yao, ZHAN Qiang, XI Xi, et al. Inverse kinematics identification of a spherical robot based on BP neural networks[C]//2011 6th IEEE Conference on Industrial Electronics and Applications. Beijing:IEEE, 2011:2114-2119.
- [7] PÉREZ-RODRÍGUEZ R, MARCANO-CEDEÑO A, COSTA Û, et al. Inverse kinematics of a 6 DOF human upper limb using ANFIS and ANN for anticipatory actuation in ADL-based physical Neurorehabilitation[J]. Expert Systems with Applications, 2012, 39(10):9612-9622.
- [8] YIN F, WANG Y N, YANG Y M. Inverse kinematics solution for robot manipulator based on neural network under joint subspace[J]. International Journal of Computers, Communications & Control, 2012, 7(3):459-472.
- [9] 王海鸣,孔凡让,赵晓伟,等.基于 BP 神经网络的机器人逆运动学新算法[J].机电一体化,2009(1):46-49.
- [10] KÖKER R. A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization[J]. Information Sciences, 2013, 222:528-543.

(下转第 22 页)