第12卷 第4期　　自动化学报　　Vol. 12, No. 4
1986年10月　　ACTA AUTOMATICA SINICA　　Oct., 1986

改进的 GMDH 方法
——原始变量保存算法

杨自厚　李宝泽　郑秉林　刘静海
（东北工学院）

摘　　要

本文在 GMDH 算法的基础上，研究了中间变量数目的合适取值问题；提出了原始变量保存的 GMDH 算法；分析了改进算法的特点．计算机仿真表明，改进的 GMDH 算法较原算法有更高的可靠性和灵活性，并可减少计算量和计算机内存容量．

六十年代末，苏联依万年柯小组运用多层神经网络原理、品种改良假说，以 K-G 多项式为基础，提出了一种复杂非线性系统的辨识方法——组合数据处理方法，简称 GMDH 方法[1]．该方法应用于复杂系统的建模、预测与控制，取得了良好的效果．

GMDH 算法主要是利用多层迭代，逐层选择，以获得系统的最终模型．它的后层运算完全是在前层运算和选择的基础上进行的．因此，最终模型的质量不仅与中间变量的选择有关，还与中间变量选择的数目有关．

本文根据对现有 GMDH 算法的分析和运算，提出一种 GMDH 改进算法——原始变量保存算法．

一、GMDH 算法的中间变量保留数

中间变量的保留数对于系统模型的建立有重要的影响．保留数目多，会使运算工作量大，并占用较多的计算机内存．但中间变量保留数过小，又会过早淘汰有用变量，使最终组合的模型质量不高．

为了说明中间变量数的取值对多层迭代结果的影响，给出如下分析结果：

设被辨识对象为线性系统，输出为 y，输入为一组独立的变量 $x_1, x_2, \cdots, x_n$，所张成的矢量空间为 $S^n = \{x_1, x_2, \cdots, x_n\}$．输出 y 在 x_i 轴上的投影记作 $Pr_{x_i}y$ 或 $Pr_{s_i}y$，且 $Pr_{s_i}y > 0$，$i = 1, 2, \cdots, n$，则输出 y 在 S^n 上的最佳逼近式为

$$\hat{y}_{s^n} = a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n, \quad \hat{y}_{s^n} \in S^n.$$

若有 $S^n \supset S^{n-1} \supset \cdots \supset S^1$，则有

本文于1984年7月26日收到．

$$Pr_{s^n}y \succ Pr_{s^{n-1}}y \succ \cdots \succ Pr_{s^1}y.$$

式中"$\succ$"表示"优于".

根据 GMDH 算法的特点和正交投影原理,以最优逼近为选择准则,使多层运算结果为上述最佳式的充要条件如下:

1) 必要条件. 最终运算结果为最佳表达式的必要条件是:第 i 层中间变量保留数 ML_i 满足如下关系:

$$ML_1 \geqslant \frac{n}{2},\ n = 2,3,4,\cdots$$

$$ML_2 \geqslant \frac{ML_1'}{2},$$

$$\vdots$$

$$ML_i \geqslant \frac{ML_{i-1}'}{2}.$$

式中

$$ML_i' = ML_{i,\min} = \min\left\{ML_i;\ ML_i \geqslant \frac{ML_{i-1}'}{2},\ i = 2,3,\cdots;\ ML_1 \geqslant \frac{n}{2}\right\}.$$

2) 充分条件. 最终运算结果为最佳表达式的充分条件是:中间变量保留数 ML_i 满足如下关系:

$$ML_1 \geqslant C_{n'}^2 + 1,\ n \geqslant 3,$$

$$ML_2 \geqslant C_{n'}^4 + 1,\ n \geqslant 5,$$

$$\vdots$$

$$ML_i \geqslant C_{n'}^{2^i} + 1;\ n \geqslant 2^i + 1.$$

式中 $n' = n - 1$,记 $ML_{i,\max} = \min\{ML_i; ML_i \geqslant C_{n'}^{2^i} + 1\}$.

必要条件是在理想条件下获得的,而充分条件是在最劣条件下获得的. 因而,为了使最终模型有较高的可信度,运算量也不过大,应适当选择中间变量的保留数.

对于非线性系统,且变量之间往往并不完全相互独立,情况比较复杂. 但是,中间变量数的适当选择仍是一个重要问题.

二、改进 GMDH 算法——原始变量保存算法

合适的中间变量保留数与输入变量的数目 n 有关. 当 n 增加时,ML 的合适数目会急剧增加,以致计算机内存不允许. 若 ML 取得过小,又会降低模型质量. 为了解决这一问题,我们改变 GMDH 算法的变量传输方式,在各层均设立一个原始变量保存区,将作用较弱的变量保存起来,直接参与各层的组合运算,避免在低层过早被淘汰,这样就可以在高层中为改进中间结果提供所需的变量信息.

原始变量保存算法的结构如图 1 所示. 这一算法的特点是:算法不改变原组合和选择方式;可以增加算法的灵活性;在中间变量保留数相同的情况下,使自组织更加充分,从而提高模型质量.

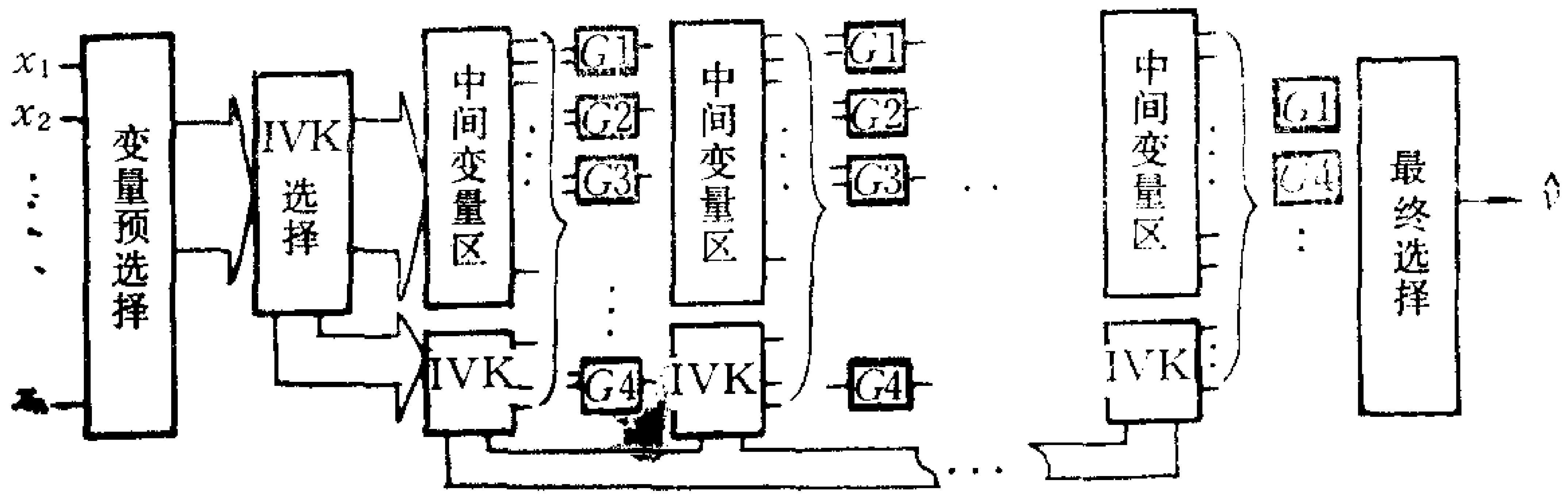

图 1　原始变量保存算法结构
IVK：原始变量保存

三、改进算法的仿真结果

为检验改进算法对非线性系统的辨识效果，本文采用如下确定型数学模型作为辨识的对象

$$y = 0.3 + x_1 + x_3 + 2x_1x_2 - x_2^2.$$

利用田村坦之的改进型 GMDH 算法[2]和本文的改进算法对该系统进行辨识.

1）数据生成．利用随机数发生器产生 x_1, x_2, x_3 的取值，y 的数据生成式为

$$y' = 0.3 + x_1 + x_3 + 2x_1x_2 - x_2^2 + e.$$

式中 e 为噪声．共产生36组数据，其中 31 组用于建模，5 组用于模型检验.

2）中间变量选择准则．利用 BIC 准则.

3）运算结果．采用田村坦之的改进型算法，如取 $ML = 3$，运算得到各层和最终表达式如下：

第一层　$z_1 = a_0 + a_1x_1x_2 - a_2x_2^2$，$z_2 = a_0 + a_1x_1$，$z_3 = a_0 + a_1x_3$；

第二层　$v_1 = z_1$，$v_2 = a_0 + a_1x_3 + a_2x_1x_2 - a_3x_2^2$，$v_3 = a_0 + a_1x_1x_2 - a_3x_2^2$；

第三层　$w_1 = w_2 = w_3 = v_2$.

最终模型　$\hat{y} = a_0 + a_1x_3 + a_2x_1x_2 - a_3x_2^2$.

可见，在第二层将 x_1 删除，在最终表达式中不包含 x_1．结构与原模型不一致.

利用本文改进算法，设置原始变量保留区 IVK，存入 x_1 和 x_3．取 $ML = 1$．运算得到各层和最终表达式为

第一层　$z_1 = 6.3184 + 2.1605x_1x_2 - 1.0642x_2^2$；

第二层　$v_1 = 3.4621 + 2.1472x_1x_2 - 1.0576x_2^2 + 0.9547x_3$；

第三层即最终模型

$$w_1 = \hat{y} = 0.3001 + 0.92857x_1 + 0.8952x_3 + 2.0133x_1x_2 - 0.9917x_2^2.$$

可见，最终表达式与原模型相当一致.

对于两种算法，各层的拟合精度和预测精度如图 2 所示.

可以看出，采用本文的改进算法，改进了高层运算的表达式；提高了拟合和预测精度；

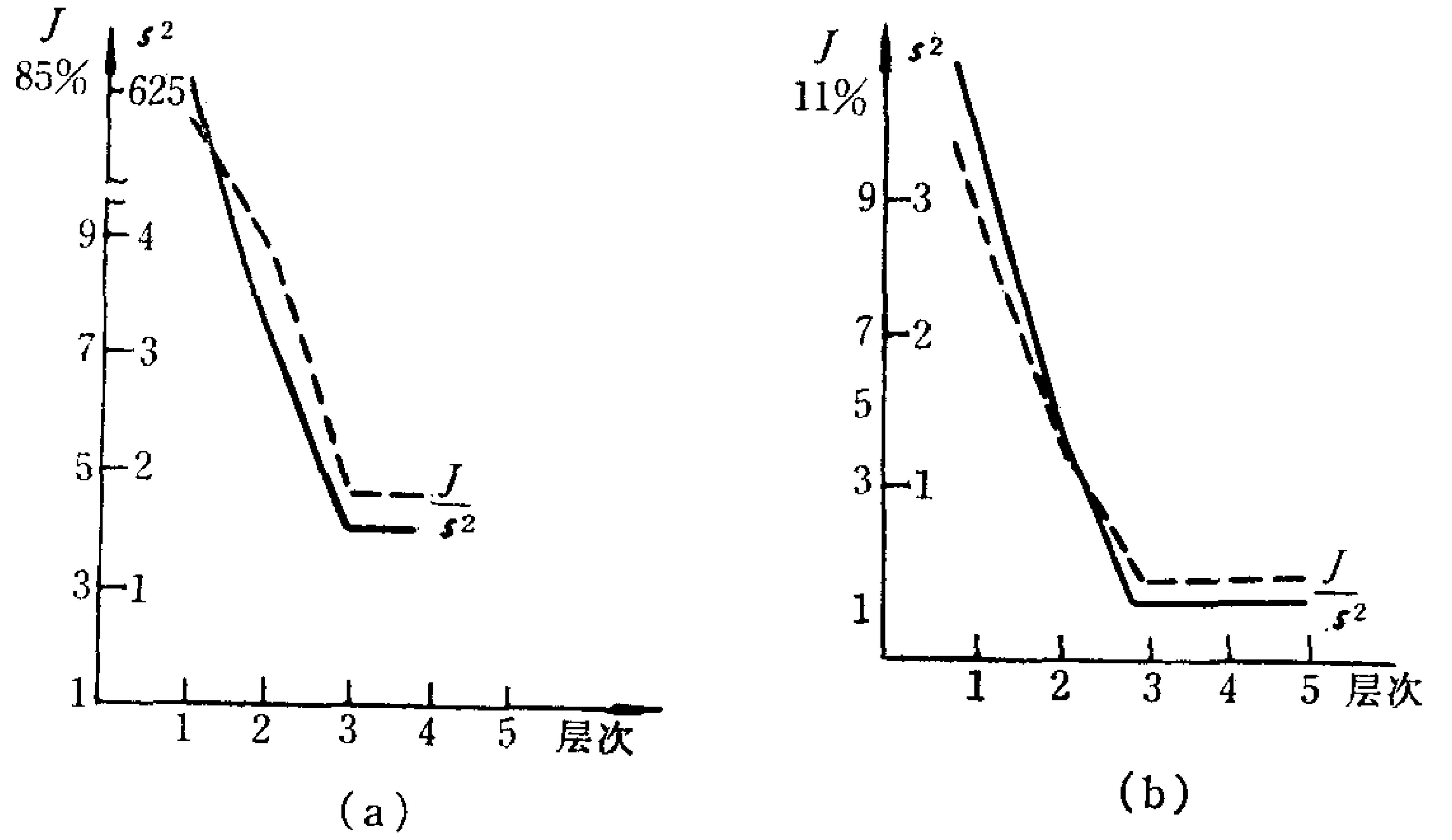

图 2 各层指标 (a) 田村算法; (b) 本文算法

模型结构与对象完全相同,参数估计也较准确.

参 考 文 献

[1] Ivakhenko, A. G., Polynomial Theory of Complex System. *IEEE Tran.* SMC-**1** (1971), 634—378.
[2] 田村坦之,近藤 正,モデル選択の評価規準に预测平方和 (PSS) を用い为改良形 GMDH, 計测自動制御学会論文集,**14**(1978),518—524.

AN IMPROVED GMDH ALGORITHM——THE INITIAL VARIABLE-KEEPING ALGORITHM

YANG ZIHOU LI BAOZE ZHENG BINGLIN LIU JINGHAI
(*Northeast University of Technology*)

ABSTRACT

Based on the self-organizing GMDH algorithm, the paper studies the proper range of the number of intermediate variables selected. An improved GMDH algorithm which keeps the initial variables is given. Features of this algorithm are analysed. Computer simulation shows that the improved GMDH algorithm is more reliable and flexible than the original algorithm, and demands much less calculation and computer memory.