

# MIMO 系统求逆的 Interactor 算法改进

胡立坤<sup>1,2</sup>, 王庆超<sup>2</sup>

(1. 广西大学 电气工程学院, 南宁 530004; 2. 哈尔滨工业大学 航天学院, 哈尔滨 150001)

**摘要:**针对多输入多输出系统求逆的 Interactor 算法出现异常中断和不能给出构造性逆的问题,提出了在实施 Interactor 算法之前对输出交换输出次序、在算法过程中若出现异常时采用恢复方法、间接求逆、引入控制输入动态的改进策略,为提高可逆性的判定功效和实现逆系统的完全解析构造提供了有效的方法。算例和仿真说明了改进的 Interactor 算法的有效性。

**关键词:**自动控制技术;多输入多输出系统;逆系统方法;Interactor 算法

**中图分类号:**TP13 **文献标识码:**A **文章编号:**1671-5497(2009)02-0494-05

## Modification of Interactor algorithm in the inverse of MIMO system

HU Li-kun<sup>1,2</sup>, WANG Qing-chao<sup>2</sup>

(1. College of Electrical Engineering, Guangxi University, Nanning 530004, China; 2. School of Astronautics, Harbin Institute of Technology, Harbin 150001, China)

**Abstract:** The Interactor algorithm for the construction of inverse of multi-input multi-output (MIMO) nonlinear systems was modified for the problems that the abnormal interrupt and no constructive inverse would happen. In the modification, output permutation before the algorithm execution, retrieval from abnormal abortion, indirect inverse solution, and introduction of improvement strategy in control dynamics were used. The proposed modification can provide an effective method for reversibility judgment and the completely analytical inverse construction. Examples and simulations were given to demonstrate validity of the proposed modified Interactor algorithm.

**Key words:** automatic control technology; multi-input multi-output systems; inverse system method; Interactor algorithm

针对逆系统方法从时域角度研究了一般非线性的右可逆性及构造<sup>[1-2]</sup>、解耦理论<sup>[3]</sup>以及系统镇定、线性化综合和状态观测等方面的基本理论和方法。求逆系统一般用 Singh 算法<sup>[4]</sup>和 Interactor 算法<sup>[5]</sup>。前一种算法对输出统一进行处理,而后一种算法则对输出分别依次处理。这两种算法最初均是针对仿射非线性系统的,后来

将其拓展<sup>[1,6]</sup>到一般 MIMO 非线性系统中。逆系统方法的应用研究发展也很迅速,在感应电机转矩与磁链的动态解耦控制<sup>[7]</sup>、无轴承永磁同步电机解耦控制<sup>[8]</sup>、机械人控制<sup>[9]</sup>、连续搅拌釜式化学反应器<sup>[10]</sup>、电力系统控制<sup>[11]</sup>、飞行控制<sup>[12]</sup>等领域都有成功应用的报道。但这些应用在不能求得解析逆时基本都是用神经网络逼近逆系统。另外,

**收稿日期:**2007-08-17.

**基金项目:**国家自然科学基金项目(60574022).

**作者简介:**胡立坤(1977-),男,博士,研究方向:航天器动力学与控制,工业测控网络.

E-mail: hlk3email@163.com

**通信作者:**王庆超(1950-),男,教授,博士生导师,研究方向:复杂大系统控制,智能控制等研究.

E-mail: wangqingchao@hit.edu.cn

文献[13]还采用支持向量机逼近逆系统;文献[14]针对 Singh 算法给出了一般性构造逆的方法。本文针对 Interactor 算法的异常中断和不能给出构造性逆的问题,对其构造逆系统的过程做了几点改进。

## 1 系统 Interactor 算法与不足

一般非线性系统的 Interactor 算法如下<sup>[6]</sup>:

Step 1 对  $y_1$  求导,可能出现两种情况:

Case 1.1 直到  $y_1^{\alpha_1}$  显含  $u_l$  ( $\exists l \in \{1, 2, \dots, m\}$ ), 即对输入  $\mathbf{u}$  满足秩条件  $\text{rank}(\partial y_1^{\alpha_1} / \partial \mathbf{u}^T) = 1$ , 记  $\alpha_1 = \rho_1, Y_1 = y_1^{\alpha_1} = y_1^{\alpha_1}(\mathbf{x}, \mathbf{u}), \gamma_1 = 1$ , 继续 Step 2。

Case 1.2 无论求多少次导数都不会出现上面的情况, 则算法结束, 意味着出现异常, 中断 Interactor 算法。

Step 2 对  $y_2$  求导, 可能出现两种情况:

Case 2.1 直到  $y_2^{\alpha_2}$  显含  $u_l$  ( $\exists l \in \{1, 2, \dots, m\}$ ), 即对输入  $\mathbf{u}$  满足  $\text{rank}(\partial y_2^{\alpha_2} / \partial \mathbf{u}^T) = 1$ , 记  $Y_2 = (Y_1 \ y_2^{\alpha_2}(\mathbf{x}, \mathbf{u}))^T$ 。分两种情况:

Case 2.2.1 如果有  $\text{rank}(\partial Y_2 / \partial \mathbf{u}^T) = 2$ , 则记  $\gamma_2 = 2, \alpha_2 = \rho_2$ , 继续 Step 3。

Case 2.2.2 若  $\text{rank}(\partial Y_2 / \partial \mathbf{u}^T) = 1$ , 则说明  $Y_1$  与  $y_2^{\alpha_2}(\mathbf{x}, \mathbf{u})$  关于  $\mathbf{u}$  是相关的。首先通过 Step 1 得到的  $Y_1$  求解相关的  $\mathbf{u}$  中分量代入到  $y_2^{\alpha_2}(\mathbf{x}, \mathbf{u})$  中, 此时有  $y_2^{\alpha_2} = y_2^{\alpha_2}(\mathbf{x}, Y_1)$ , 然后继续对  $y_2^{\alpha_2}$  求导, 在正常情况下可以得到  $\hat{\rho}_2 > 0$  使  $y_2^{\alpha_2 + \hat{\rho}_2}$  对显含  $u_l$  ( $\exists l \in \{1, 2, \dots, m\}$ ),  $\text{rank}(\partial y_2^{\alpha_2 + \hat{\rho}_2} / \partial \mathbf{u}^T) = 1$ , 有  $y_2^{(\alpha_2 + \hat{\rho}_2)} = y_2^{(\alpha_2 + \hat{\rho}_2)}(\mathbf{x}, y_1^{\alpha_1 + \hat{\rho}_2}, \mathbf{u})$ , 且可记  $Y_2 = (Y_1 \ y_2^{\alpha_2 + \hat{\rho}_2}(\mathbf{x}, y_1^{\alpha_1 + \hat{\rho}_2}, \mathbf{u}))^T$ , 否则算法非正常结束。

同样正常情况下, 会出现下面的两种情况:

Case 2.2.2.1 如果有  $\text{rank}(\partial Y_2 / \partial \mathbf{u}^T) = 2$ , 则记  $\gamma_2 = 2, \alpha_2 = \rho_2 + \hat{\rho}_2$ , 继续 Step 3。

Case 2.2.2.2 若  $\text{rank}(\partial Y_2 / \partial \mathbf{u}^T) = 1$ , 则说明  $Y_1$  与  $y_2^{(\alpha_2 + \hat{\rho}_2)}$  关于  $\mathbf{u}$  仍是相关的, 首先通过 Step 1 求解相关的  $\mathbf{u}$  中分量, 代入到  $y_2^{(\alpha_2 + \hat{\rho}_2)}(\mathbf{x}, y_1^{(\alpha_1 + \hat{\rho}_2)}, \mathbf{u})$  中, 此时有  $y_2^{(\alpha_2 + \hat{\rho}_2)} = y_2^{(\alpha_2 + \hat{\rho}_2)}(\mathbf{x}, y_1^{(\alpha_1 + \hat{\rho}_2)}, y_1^{\alpha_1})$ , 然后继续对  $y_2^{(\alpha_2 + \hat{\rho}_2)}$  求导, 重复 Step 2 前面的过程。

如果在此步算法正常结束, 必存在  $\alpha \in \mathbf{N}, Y_2 = (Y_1, y_2^{\alpha}(\mathbf{x}, \{y_1^{\hat{\rho}_1} \mid i_1 = \alpha_1, \dots, \alpha_1 + \alpha_2\}, \mathbf{u}))^T$  使  $\text{rank}(\partial Y_2 / \partial \mathbf{u}^T) = 2$ , 同时记  $\gamma_2 = 2$ ; 否则异常中断 Interactor 算法。

Case 2.2 无论求多少次导数都不会出现 Case 2.1, 则算法结束, 意味着出现异常中断。重复 Step 2 的过程, 假设算法从 Step 1 到 Step  $r$  都是正常结束, 必有秩的序列  $\{\gamma_1, \gamma_2, \dots, \gamma_r\}$  和非负相对阶向量  $(\alpha_1, \alpha_2, \dots, \alpha_r)^T$ , 即满足  $\gamma_r = \text{rank}(\partial Y_r / \partial \mathbf{u}^T) = r$ , 其中

$$\mathbf{Y}_r = \begin{pmatrix} y_1^{\alpha_1} \\ y_2^{\alpha_2} \\ \vdots \\ y_r^{\alpha_r} \end{pmatrix} =$$

$$\begin{pmatrix} y_1^{\alpha_1}(\mathbf{x}, \mathbf{u}) \\ y_2^{\alpha_2}(\mathbf{x}, \{y_1^{\hat{\rho}_1} \mid i_1 = \alpha_1, \dots, \alpha_1 + \alpha_2\}, \mathbf{u}) \\ \vdots \\ y_r^{\alpha_r}(\mathbf{x}, \{y_j^{\hat{\rho}_j} \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_r, \\ j = 1, 2, \dots, r-1\}, \mathbf{u}) \end{pmatrix}$$

$$\text{令} \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_r \end{pmatrix} =$$

$$\begin{pmatrix} y_1^{\alpha_1}(\mathbf{x}, \mathbf{u}) - y_1^{\alpha_1} \\ y_2^{\alpha_2}(\mathbf{x}, \{y_1^{\hat{\rho}_1} \mid i_1 = \alpha_1, \dots, \alpha_1 + \alpha_2\}, \mathbf{u}) - y_2^{\alpha_2} \\ \vdots \\ y_r^{\alpha_r}(\mathbf{x}, \{y_j^{\hat{\rho}_j} \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_r, \\ j = 1, 2, \dots, r-1\}, \mathbf{u}) - y_r^{\alpha_r} \end{pmatrix}$$

则  $(\eta_1, \eta_2, \dots, \eta_r)^T = \mathbf{0}$ 。

在该算法正常结束的情况下, 令  $\bar{\mathbf{y}} \triangleq (\{y_j^{\hat{\rho}_j} \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_r, j = r\})^T, r = 1, 2, \dots, r$ , 并且在  $t = t_0$  处有  $\bar{\mathbf{y}}(t_0) = \bar{\mathbf{y}}_0, \mathbf{x}(t_0) = \mathbf{x}_0, \mathbf{u}(t_0) = \mathbf{u}_0$ , 在此基础上可以给出基于该算法的 MIMO 非线性系统的正则点、右向量相对阶、向量本性阶、本性秩数值定义<sup>[11]</sup>。该算法存在以下问题:

(1) 基于 Interactor 算法求逆异常中断时就意味着不能使用逆系统对其控制, 事实上并非这样, 即可控性指数虽然与输出不相等, 但系统可能是仍可镇定的, 所以还是有必要确定 MIMO 非线性系统可控性指数。

(2) 标准 Interactor 算法并非完全构造性的, 虽然可以判定逆是存在的, 但缺乏有效的求逆方法。

## 2 Interactor 算法的改进

改进 1(交换输出次序): 为减少求导次数, 将输出表达式中显含不同  $\mathbf{u}$  分量的输出放在前面,

即在 Step 1 前加一步 Step 0 调整输出顺序。

Step 0 将显含不同  $u$  分量的输出放在前面, 设新的输出为  $y' = (y'_1, y'_2, \dots, y'_r)^T$ , 通过交换矩阵  $T$  完成交换

$$T = (t_{ij})_{r \times r}, t_{ij} = \begin{cases} 1, & \text{if } y'_i = y_j \\ 0, & \text{其他} \end{cases}, y' = Ty$$

需要注意的是, 这一步并不是说明交换到前面的那些含  $u$  分量的输出均可以求出相对阶。当然, 输出顺序的改变会引起对应的相对阶和本性阶的改变, 但不改变相对阶之和, 也不改变算法过程的约束。

**改进 2**(从异常中断恢复): 从算法中可以看出, 异常中断均是由于对某输出求导一直不能出现新的输入分量, 但是并不意味着对后面的输出分量求导不会出现新的输入分量。在这种情况下, 可以将该分量交换到最后继续算法过程, 而将余下的分量前移, 直到剩余的输出分量均是被交换过的分量, 为此需标识已交换过的分量。这样持续到不会出现新的  $u$  分量时, 算法结束。由此得到的步数便是可控性指数。

**改进 3**(整体处理): 当前面的算法步中相关分量组成因式以一个含  $u$  中多个元素的表达式形式给出时, 可以将其看成一个整体解出, 代入当前算法步中继续计算, 若解不出来或不易计算, 可以采用改进 4 间接求逆的方法。

**改进 4**(改进中间过程求逆方法——间接求逆): 假设第  $k$  步在计算过程中需将第  $k-1$  步计算  $u$  的分量  $u_{k-1}$  代入, 由于在 Step  $k$  步中, 若不进行代换意味着出现  $u$  的分量的导数, 基于此若能从 Step  $k-1$  步求出该导数代入就更方便了, 故而设 Step  $k-1$  完成后的  $Y_{k-1}$  表达式变为

$$\bar{\eta}_{k-1}(x, \{y_j^i \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_{k-1}, j = \{1, 2, \dots, k-1\}\}, u_{k-1}) = 0 \quad (1)$$

对上式求导, 可得

$$(\partial \bar{\eta}_{k-1} / \partial x^T) \dot{x} + (\partial \bar{\eta}_{k-1} / \partial u^T) \dot{u} + (\partial \bar{\eta}_{k-1} / \partial y^T) \dot{y}'' = 0 \quad (2)$$

式中:  $\bar{y}'' \triangleq (\{y_j^i \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_{k-1}, j = k-1\})^T$ 。

为方便计算, 这里只考虑与前  $k-1$  个方程所含的  $k-1$  个  $u$  的分量  $u_{k-1}$  (角标  $k-1$  表示  $u_{k-1}$  中的元素个数), 所以  $\partial \bar{\eta}_{k-1} / \partial u_{k-1}^T$  是方的, 且由 Interactor 算法可知  $\det(\partial \bar{\eta}_{k-1} / \partial u_{k-1}^T) \neq 0$ , 所以可由式(2)求出  $\dot{u}_{k-1}$ , 即

$$\dot{u}_{k-1} = -(\partial \bar{\eta}_{k-1} / \partial u_{k-1}^T)^{-1} (\partial \bar{\eta}_{k-1} / \partial x^T) \dot{x} - (\partial \bar{\eta}_{k-1} / \partial u_{k-1}^T)^{-1} (\partial \bar{\eta}_{k-1} / \partial y^T) \dot{y}'' \quad (3)$$

将上式中与 Step  $k$  步中相关的分量导数代入后, 继续 Interactor 算法。

通过简单的整体替换可以应用于改进 3 的整体处理。在具体实施算法时可以将直接求逆与间接求逆方法交叉使用, 间接求逆消除了求逆瓶颈。

**改进 5**(逆系统构造的改进——引入控制输入动态): 假设按 Interactor 算法进行了  $r$  步后, 得到系统的右相对阶向量为  $\alpha = (\alpha_1 \ \alpha_2 \ \dots \ \alpha_r)^T$ , 若  $m > r$ , 则按可逆性证明<sup>[11]</sup>构造一个扩展函数向量  $\tilde{\eta} = (\tilde{\eta}_1 \ \tilde{\eta}_2 \ \dots \ \tilde{\eta}_{m-r})^T$ , 其中  $\tilde{\eta}_k$  元素表示为  $\tilde{\eta}_k(x, \{y_j^i \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_r, j = r\}, u) = \text{const}, \kappa = 1, 2, \dots, m-r$ , 使  $\text{rank}(\partial(\eta^T, \tilde{\eta}^T)^T / \partial u^T) = m$ , 并形成  $(\eta^T, \tilde{\eta}^T)^T = (0^T, C^T)^T$ , 对其求导可得

$$(\partial(\eta^T, \tilde{\eta}^T)^T / \partial x^T) \dot{x} + (\partial(\eta^T, \tilde{\eta}^T)^T / \partial u^T) \dot{u} + (\partial(\eta^T, \tilde{\eta}^T)^T / \partial y^T) \dot{y} = 0 \quad (4)$$

式中:  $\bar{y} \triangleq (\{y_j^i \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_r, j = r\})^T$ 。

由于  $\partial(\eta^T, \tilde{\eta}^T)^T / \partial u^T$  是方的, 且  $\det(\partial(\eta^T, \tilde{\eta}^T)^T / \partial u^T) \neq 0$ , 于是求解  $\dot{u}$

$$\dot{u} = (\partial(\eta^T, \tilde{\eta}^T)^T / \partial u^T)^{-1} ((\partial(\eta^T, \tilde{\eta}^T)^T / \partial x^T) f(x, u) + (\partial(\eta^T, \tilde{\eta}^T)^T / \partial y^T) \dot{y}) = \Gamma(x, u, \{y_j^i \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_r + 1, j = r\}) \quad (5)$$

根据式(5)引入关于系统控制量的动态补偿构造如下的逆系统

$$\dot{w} = \Gamma(x, u, \{y_j^i \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_r + 1, j = r\}) \quad (6)$$
$$u = w \quad (7)$$

引入这种动态补偿后, 可以构造任何形式的解析非线性系统的逆系统, 但同时增加了  $m$  阶的动态。如果可能解出  $u$  就不需引入系统控制量的动态补偿, 只需通过原始的静、动态解耦即可。

在一些情况下, 有一部分输入可以显式地由状态和其他输入表示, 由此可得到降阶的补偿器。假设这部分输入为  $u_a$  (个数为  $a$ ), 其余输入为  $u_b$  (个数为  $b$ ), 即  $u = (u_a^T, u_b^T)^T, a + b = m$ 。为方便书写, 令  $\eta$  为扩展后的函数向量, 从中选择出可以显式解出  $u_a$  的  $a$  个方程  $\eta_a = 0$ , 解之得

$$\mathbf{u}_a = \boldsymbol{\eta}_a^{-1}(\mathbf{x}, \{y_j^i \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_r, j = r\} \mathbf{u}_b) \quad (8)$$

对其余的  $b$  个方程  $\boldsymbol{\eta}_b = 0$ , 按式(4)两边求导可得

$$\left( \frac{\partial \boldsymbol{\eta}_b}{\partial \mathbf{x}^T} + \frac{\partial \boldsymbol{\eta}_b}{\partial \mathbf{u}_a^T} \frac{\partial \mathbf{u}_a}{\partial \mathbf{x}^T} \right) \dot{\mathbf{x}} + \left( \frac{\partial \boldsymbol{\eta}_b}{\partial \mathbf{u}_b^T} + \frac{\partial \boldsymbol{\eta}_b}{\partial \mathbf{u}_a^T} \frac{\partial \mathbf{u}_a}{\partial \mathbf{u}_b^T} \right) \dot{\mathbf{u}}_b + \left( \frac{\partial \boldsymbol{\eta}_b}{\partial \mathbf{y}^T} + \frac{\partial \boldsymbol{\eta}_b}{\partial \mathbf{u}_a^T} \frac{\partial \mathbf{u}_a}{\partial \mathbf{y}^T} \right) \dot{\mathbf{y}} = 0 \quad (9)$$

显然, 式(9)中  $\dot{\mathbf{u}}_b$  的系数满秩, 即使  $\boldsymbol{\eta}$  为扩展以前的函数向量,  $\dot{\mathbf{u}}_b$  也是行满秩的(相当于欠约束方程组), 所以上述方程中  $\dot{\mathbf{u}}_b$  可解, 令其为

$$\dot{\mathbf{u}}_b = \boldsymbol{\Gamma}_b(\mathbf{x}, \mathbf{u}_b, \{y_j^i \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_r + 1, j = r\}) \quad (10)$$

于是, 得到  $b(\leq m)$  阶的动态补偿逆系统

$$\dot{\mathbf{w}}_b = \boldsymbol{\Gamma}(\mathbf{x}, \mathbf{w}_b, \{y_j^i \mid i_j = \alpha_j, \dots, \alpha_j + \alpha_{j+1} + \dots + \alpha_r + 1, j = r\}) \quad (11)$$

$$\mathbf{u} = (\mathbf{u}_a^T \mathbf{w}_b^T)^T \quad (12)$$

### 3 算 例

**例 1** 考虑如下的一般非线性系统, 系统输入  $\mathbf{u} = (u_1 \ u_2 \ u_3)^T$ , 输出  $\mathbf{y} = (y_1 \ y_2 \ y_3)^T$ ,

$$\begin{cases} \dot{x}_1 = x_2 + u_1 \\ \dot{x}_2 = x_3 + u_2 \\ \dot{x}_3 = -x_3 \\ \dot{x}_4 = x_2 + x_3 u_3 \end{cases}, \begin{cases} y_1 = x_1 \\ y_2 = x_3 \\ y_3 = x_4 \end{cases}$$

由于输出中没有显含  $\mathbf{u}$  的分量的项, 所以不用 Step 0 步。按 Interactor 算法处理此系统, 在 Step 2 对  $y_2$  无论求多少次导数均不会显含  $\mathbf{u}$  的分量, 所以改变余下输出顺序,  $y_2 = x_4, y_3 = x_3$ 。继续对新的  $y_2$  求导,  $\dot{y}_2 = x_2 + x_3 u_3$ , 已显含  $u_3, Y_2 = (Y_1 \ \dot{y}_2)^T$ , 所以  $\alpha_2 = 1, \gamma_2 = 2$ 。

由于剩下的  $y_3$  为交换出去的, 所以算法结束, 可控性指数为 2。同时也可看出不可控的输出是稳定的, 这也是需采用改进 2 的原因。

**例 2** 考虑如下的一般非线性系统, 系统输入  $\mathbf{u} = (u_1 \ u_2 \ u_3)^T$ , 输出  $\mathbf{y} = (y_1 \ y_2 \ y_3)^T$ ,

$$\begin{cases} \dot{x}_1 = x_1 u_1 + u_3 e^{u_1} \\ \dot{x}_2 = x_4 + u_1 \\ \dot{x}_3 = x_1 e^{u_1} + u_2 \\ \dot{x}_4 = x_3 + x_4 u_2^2 \\ \dot{x}_5 = x_1 + x_4 + u_1^2 - u_1 \end{cases}, \begin{cases} y_1 = x_1 u_1 e^{u_1} \\ y_2 = x_2 \\ y_3 = x_5 \end{cases}$$

按 Interactor 算法处理此系统, 在计算第 2 步时出现  $\dot{u}_1$ , 便可由第 1 步求得的  $\boldsymbol{\eta}_1 = x_1 u_1 e^{u_1} -$

$y_1 = 0$  两边求导得  $(\partial \boldsymbol{\eta}_1 / \partial x_1) \dot{x}_1 + (\partial \boldsymbol{\eta}_1 / \partial u_1) \dot{u}_1 + (\partial \boldsymbol{\eta}_1 / \partial y_1) \dot{y}_1 = 0, \dot{u}_1 = (\dot{y}_1 - u_1 e^{u_1} (x_1 u_1 + u_3 e^{u_1})) / (e^{u_1} + u_1 e^{u_1})$ , 于是  $\ddot{y}_2 = x_3 + x_4 u_2^2 + (\dot{y}_1 - u_1 e^{u_1} (x_1 u_1 + u_3 e^{u_1})) / (e^{u_1} + u_1 e^{u_1})$ , 可继续向下计算。

**例 3** 考虑如下的一般非线性系统, 系统的输入  $\mathbf{u} = (u_1 \ u_2 \ u_3)^T$ , 输出为  $\mathbf{y} = (y_1 \ y_2)^T$ ,

$$\begin{cases} \dot{x}_1 = x_2 + u_1 \\ \dot{x}_2 = x_3 + u_2 \\ \dot{x}_3 = x_3 \\ \dot{x}_4 = x_2 + u_3 e^{u_3} \end{cases}, \begin{cases} y_1 = x_1 \\ y_2 = x_4 \end{cases}$$

按改进的 Interactor 算法处理此系统后得到  $\boldsymbol{\alpha} = \boldsymbol{\beta} = (1 \ 1)^T$ 。并有  $\boldsymbol{\eta}_1 = x_2 + u_1 - \dot{y}_1 = 0, \boldsymbol{\eta}_2 = x_2 + u_3 e^{u_3} - \dot{y}_2 = 0$ 。构造  $\tilde{\boldsymbol{\eta}}_1 = x_2 + u_2 = 0$ , 这时  $\det(\partial \boldsymbol{\eta}^T, \tilde{\boldsymbol{\eta}}^T) / \partial \mathbf{u}^T = u_3 e^{u_3} + e^{u_3}$ , 需满足  $u_3 e^{u_3} + e^{u_3} \neq 0$ 。显然  $u_1 = \dot{y}_1 - x_2, u_2 = -x_2$ , 而  $\dot{u}_3 = -(\dot{x}_2 - \ddot{y}_2) / (u_3 e^{u_3} + e^{u_3})$ 。于是, 得到 1 阶动态补偿逆系统  $\dot{w}_1 = -(\dot{x}_2 - \ddot{y}_2) / (w_1 e^{w_1} + e^{w_1}), \mathbf{u} = (u_1 \ u_2 \ w_1)$ 。

将所求的逆系统串接于原系统之前, 形成伪线性系统, 解耦数值仿真如图 1 所示。需要注意的是, 对输出  $y_2$  的解耦在引入动态补偿后变成双积分形式。

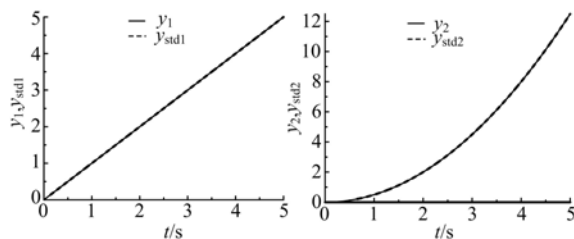


图 1 例 3 的解耦效果

Fig. 1 Decoupling effect of example 3

### 4 结束语

本文在分析 Interactor 算法过程基础上, 提出了 Interactor 算法的几点改进。这些改进为提高可逆性的判定功效和实现逆系统的完全解析构造提供了有效的方法。当且仅当不可控部分是稳定的, 且可控制部分的可控性指数与输出个数相等, 则得到的逆系统就是有效的。算例进一步解释了这些改进, 对解耦效果的仿真验证了解析构造的可行性。

## 参考文献:

- [1] Li C W, Feng Y K. Functional reproducibility of general multivariable nonlinear systems[J]. Int J Control, 1987, 45(1): 255-268.
- [2] Respondek W. On local right-invertibility of nonlinear control systems[J]. Control Theory and Advanced Technology, 1988, 4(3): 325-348.
- [3] 戴先中, 刘军, 冯纯伯. 连续非线性系统的神经网络 $\alpha$ 阶逆系统控制方法[J]. 自动化学报, 1998, 24(4): 463-468.  
Dai Xian-zhong, Liu Jun, Feng Chun-bo. Neural network  $\alpha$ -th-order inverse-system control method for nonlinear continuous systems[J]. Acta Automatica Sinica, 1998, 24(4):463-468.
- [4] Singh S N. A modified algorithm for invertibility in nonlinear systems[J]. IEEE Automat Cotro, 1981, 26(4): 595-598.
- [5] Benedetto M D D, Glumineau A, Moog C H. The nonlinear interactor and its application to input-output decoupling[J]. IEEE Automat Cotro, 1994, 39(6): 1246-1250.
- [6] Wang W C, Dai X Z. An interactor algorithm for invertibility in general nonlinear systems[C]// Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, China, 2004.
- [7] 杨立永, 李华德, 王久和. 基于逆系统理论的感应电动机控制策略[J]. 辽宁工程技术大学学报, 2005, 24(5): 706-708.  
Yang Li-yong, Li Hua-de, Wang Jiu-he. Induction motor control based on inverse system theory[J]. Journal of Liaoning Technical University, 2005, 24(5):706-708.
- [8] 费德成, 朱焯秋. 基于逆系统理论的无轴承永磁同步电机解耦控制研究[J]. 中国工程科学, 2005, 7(11): 48-54.  
Fei De-cheng, Zhu Huang-qiu. Study on decoupling control of bearingless permanent magnet synchronous motors based on inverse system theory[J]. Engineering Science, 2005, 7(11):48-54.
- [9] 戴先中, 孟正大, 沈建强, 等. 神经网络 $\alpha$ 阶逆系统控制方法在机器人解耦控制中的应用[J]. 机器人, 2001, 23(4): 363-367.  
Dai Xian-zhong, Meng Zheng-da, Shen Jian-qiang, et al. Robot decoupling control based on ANN  $\alpha$ -th-order inverse system method[J]. Robot, 2001, 23(4):363-367.
- [10] 李春文, 张平, 冯元琨. 一种基于逆系统方法的化学反应器改进控制方案[J]. 控制与决策, 1998, 13(5): 577-580.  
Li Chun-wen, Zhang Ping, Feng Yuan-kun. An improved control approach of a chemical Reactor based on the inverse system method[J]. Control and Decision, 1998, 13(5):577-580.
- [11] 张腾. 神经网络逆系统方法及其在电力系统控制中的应用[D]. 南京:东南大学自动化学院, 2002.  
Zhang Teng. Neural network inverse system and its application in power system control[D]. Nanjing: School of Automation Southeast University, 2002.
- [12] 张怡哲, 邓建华. 逆系统方法在飞行控制律设计中的工程应用[J]. 西北工业大学学报, 2006, 24(1): 35-39.  
Zhang Yi-zhe, Deng Jian-hua. Nonlinear control law design for high angle of attack flight[J]. Journal of Northwestern Polytechnical University, 2006, 24(1):35-39.
- [13] 宋夫华, 李平. 基于支持向量机阶逆系统方法的非线性内模控制[J]. 自动化学报, 2007, 33(7): 778-781.  
Song Fu-hua, Li Ping. Nonlinear internal model control based on support vector machine  $\alpha$ -th-order inverse system method[J]. Acta Automatica Sinica, 2007, 33(7): 778-781.
- [14] 吴热冰, 李春文. 一般非线性系统的构造性逆系统方法[J]. 控制理论与应用, 2003, 20(3): 345-350.  
Wu Re-bing, Li Chun-wen. Constructive inverse system method for general nonlinear systems[J]. Control Theory and Applications, 2003, 20(3):345-350.