



南開大學

基于麦夸特算法的混沌二级摆参数辨识

作	者	乔梓韶
邮	箱	<u>2111227@mail.nankai.edu.cn</u>
学	校	<u>南开大学人工智能学院</u>

2024 年 5 月 29 日

基于麦夸特算法的混沌二级摆参数辨识

摘 要

《建模与辨识》是自动化专业学生必修的一部分知识体系，其内容包括线性静态系统的最小二乘辨识、线性动态系统的最小二乘辨识、相关噪声情形下的线性系统模型辨识、模型阶的辨识、非线性系统辨识、神经网络模型辨识、遗传算法及其在辨识中的应用等等。

本项目用于记录作者对“结构已知非线性系统参数辨识”内容做实践验证，选择使用麦夸特算法，针对典型非线性系统——混沌二级摆，以复习笔记的形式，主要包含理论知识展示、编程实现流程，实现效果等部分，方便笔者未来作为参考。

关键词：建模与辨识，混沌二级摆，麦夸特算法

第 1 章 迭代思想

假设有一非线性系统, $y = f(u, \theta)$, 其中 $f(\cdot)$ 已知, 即结构已知的非线性系统, 现考虑通过已知的输入输出数据对系统参数进行辨识。

相较于一般线性系统, 非线性系统往往存在多平衡点, 且如构造目标函数形如 $J(\theta) = \sum [y(k) - f(u(k), \theta)]^2$, 求一阶偏导数并不能将辨识转化为凸优化问题, 同时一阶偏导等于零的公式往往还伴随着没有解析解、解个数未知、局部最优解等问题, 故现引入迭代思想应对非线性系统参数辨识。

1.1 基本思想

对于一结构已知的非线性系统, 给定一组参数初值 $\theta^{(0)}$, 每次迭代对 θ 做一 $\Delta\theta$ 的修正, 经过历次迭代, 得到 $\theta^{(1)}$ 、... $\theta^{(n)}$, 且对任意 n 都满足 $J(\theta^{(n)}) > J(\theta^{(n+1)})$ 。

具体到每一次迭代, 若 $\theta^{(i-1)} \rightarrow \theta^{(i)}$, 使得 $J(\theta^{(i-1)}) > J(\theta^{(i)})$, 则我们称第 i 次迭代是可接受的。

1.2 一般步骤

- 确定初值 $\theta^{(0)}$
- 确定迭代方向 v_i
- 确定迭代步长 ρ_i
- 判断终止条件

第 2 章 麦夸特算法

2.1 介绍

麦夸特算法 Levenberg-Marquardt 是解决结构已知的非线性系统辨识的有效方法, 结合了梯度下降法和牛顿-拉弗森算法的优势, 高效地解决参数辨识问题。

实际上麦夸特算法并不是一个广为流传的参数辨识方法, 其主要是在化学动力学领域应用于解决反应动力学的参数估计问题。该算法通过线性化反应速率方程然后求解它们来拟合实验数据, 但这种方法对问题的适用性有一定的限制 (算法缺点):

1. 问题的线性化: 麦夸特算法在处理经过线性化的模型时效果最好。对于不能有效线性化的复杂非线性系统, 该方法可能不适用或效率较低。
2. 初值依赖: 类似于其他迭代优化算法, 麦夸特算法可能依赖于合理的初始参数估计。错误的初始值可能导致算法收敛到非最优解。(这一点在实验中明显地体现)
3. 局部最优解: 麦夸特算法可能会找到局部最优而不是全局最优参数集, 尤其是在目标函数存在多个极小值时, 当系统非线性较强时, 往往会存在多个局部最优解, 这时算法对初值选择的依赖会进一步扩大。

2.2 算法实现

问题提出: 针对一非线性系统, $y = f(u, \theta)$, 其中 $f(\cdot)$ 已知, 即结构已知的非线性系统, 现考虑通过已知的输入输出数据对系统参数进行辨识。

2.2.1 原理介绍

基于该系统, 令偏差 $e(k) = y(k) - f(u(k), \theta)$, 设计目标函数:

$$J(\theta) = \sum [y(k) - f(u(k), \theta)]^2$$

则可以得到参数估计 $\hat{\theta} = \operatorname{argmin} J(\theta)$ 将式中 $y = f(u, \theta)$ 泰勒展开为线性函数 (在 $\theta^{(i)}$ 附近):

$$f(u, \theta) = f(u, \theta^{(i)}) + \left(\frac{\partial f}{\partial \theta}\right)^T|_{\theta=\theta^{(i)}} (\theta - \theta^{(i)})$$

则 $J(\theta)$ 变为线性函数:

$$J(\theta) = \sum [y(k) - f(u(k), \theta)]^2 = \sum [y(k) - f(u, \theta^{(i)}) + \left(\frac{\partial f}{\partial \theta}\right)^T|_{\theta=\theta^{(i)}} (\theta - \theta^{(i)})]^2$$

对 $J(\theta)$ 求一阶偏导数 $\frac{\partial J}{\partial \theta}$ 并令其等于 0:

$$\frac{\partial J}{\partial \theta} = -2 \sum \left(\frac{\partial f}{\partial \theta} \right) |_{\theta=\theta^{(i)}} [y(k) - f(u, \theta^{(i)})] - \left(\frac{\partial f}{\partial \theta} \right)^T |_{\theta=\theta^{(i)}} (\theta - \theta^{(i)}) = 0$$

其中取梯度 G_i :

$$G_i = \frac{\partial J}{\partial \theta} |_{\theta=\theta^{(i)}} = -2 \sum \left(\frac{\partial f}{\partial \theta} \right) |_{\theta=\theta^{(i)}} [y(k) - f(u, \theta^{(i)})]$$

化简第一式即有:

$$\sum \left(\frac{\partial f}{\partial \theta} \right) |_{\theta=\theta^{(i)}} [y(k) - f(u, \theta^{(i)})] - \sum \left(\frac{\partial f}{\partial \theta} \right) |_{\theta=\theta^{(i)}} \left(\frac{\partial f}{\partial \theta} \right)^T |_{\theta=\theta^{(i)}} (\theta - \theta^{(i)}) = 0$$

遂得到 $\theta^{(i+1)}$:

$$\theta^{(i+1)} = \theta^{(i)} + \left[\sum \left(\frac{\partial f}{\partial \theta} \right) |_{\theta=\theta^{(i)}} \left(\frac{\partial f}{\partial \theta} \right)^T |_{\theta=\theta^{(i)}} \right]^{-1} \sum \left(\frac{\partial f}{\partial \theta} \right) |_{\theta=\theta^{(i)}} [y(k) - f(u, \theta^{(i)})]$$

令 $A_i = \sum \left(\frac{\partial f}{\partial \theta} \right) |_{\theta=\theta^{(i)}} \left(\frac{\partial f}{\partial \theta} \right)^T |_{\theta=\theta^{(i)}}$, 则可将上式化简为:

$$\theta^{(i+1)} = \theta^{(i)} - \frac{1}{2} A_i^{-1} G_i$$

麦夸特算法对 $\theta^{(i+1)}$ 做了一定修正以保证方向 $v_i = R_i Q_i$ 是可接受的, 即:

$$\theta^{(i+1)} = \theta^{(i)} - v = \theta^{(i)} - \frac{1}{2} (A_i + \lambda I)^{-1} G_i$$

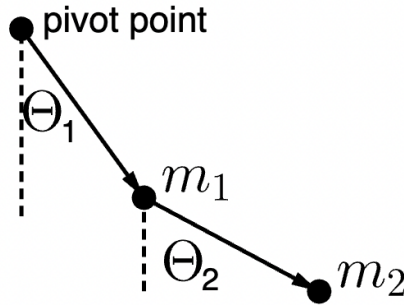
并且在迭代过程中调整 λ 的值以帮助更快收敛。

2.2.2 具体步骤

1. 猜测初始值 $\theta^{(1)}$, 计算 $J_1 = J(\theta^{(1)})$; 令 $\lambda_0 = 1, i = 1$
2. 计算 $A_i = \sum \left(\frac{\partial f}{\partial \theta} \right) |_{\theta=\theta^{(i)}} \left(\frac{\partial f}{\partial \theta} \right)^T |_{\theta=\theta^{(i)}}$, $Q_i = \frac{1}{2} G_i = \sum \left(\frac{\partial f}{\partial \theta} \right) |_{\theta=\theta^{(i)}} [y(k) - f(u, \theta^{(i)})]$
3. 计算 $v = -(A_i + (\lambda_{i-1}/10)I)^{-1} Q_i$, $\hat{\theta} = \theta^{(i)}$, $\hat{J} = J(\hat{\theta})$
4. 若 $\hat{J} < J_i$, 则 $\theta^{(i+1)} = \hat{\theta}$, 并跳转步骤 (6); 否则继续;
5. 用 $10 * \lambda_{i-1}$ 代替 λ_i 转步骤 (3), 在可行的方向上加速收敛;
6. 若 $\max_{1 \leq j \leq n} \left\{ \frac{\Delta \theta_j^{(i)}}{\theta_j^{(i)}} \right\} < \varepsilon$, 则迭代停止, 达成终止条件, 取 $\theta^{(i+1)}$ 为最中结果, 否则置 $i = i + 1$, 回到步骤 (2)。

第3章 混沌二级摆模型

混沌二级摆，也称为双摆，是一个经典的动力学系统，它展示出复杂的动态行为，包括混沌。混沌理论研究动态系统的非线性和不可预测的行为，而混沌二级摆是其中一个典型的例子。这种摆由两个质量分别挂在两根摆杆的末端，其中第一个摆杆固定在一个轴点上，而第二个摆杆则连接在第一个摆的末端。



3.1 动力学方程

每个小棍的转动惯量为：

$$I_1 = \frac{1}{12} m_1 l_1^2, \quad I_2 = \frac{1}{12} m_2 l_2^2$$

小棍一质心位置为：

$$x_1 = \frac{l_1}{2} \sin(\theta_1)$$

$$y_1 = -\frac{l_1}{2} \cos(\theta_1)$$

小棍二质心位置为：

$$x_2 = l_1 \sin(\theta) + \frac{l_2}{2} \sin(\theta_2)$$

$$y_2 = -l_1 \cos(\theta) - \frac{l_2}{2} \cos(\theta_2)$$

小棍一的质心速度为：

$$dx_1 = \frac{l_1}{2} \cos(\theta_1) \dot{\theta}_1$$

$$dy_1 = \frac{l_1}{2} \sin(\theta_1) \dot{\theta}_1$$

小棍二的质心速度为:

$$dx_2 = l_1 \cos(\theta_1) \dot{\theta}_1 + \frac{l_2}{2} \cos(\theta_2) \dot{\theta}_2$$

$$dy_2 = -l_1 \sin(\theta_1) \dot{\theta}_1 - \frac{l_2}{2} \sin(\theta_2) \dot{\theta}_2$$

拉格朗日量:

$$L = \frac{1}{2} m_1 (dx_1^2 + dy_1^2) + \frac{1}{2} m_2 (dx_2^2 + dy_2^2) + \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 - m_1 g y_1 - m_2 g y_2$$

随后可解方程组:

$$p_1 = \frac{\partial L}{\partial \dot{\theta}_1}$$

$$p_2 = \frac{\partial L}{\partial \dot{\theta}_2}$$

得到 $\dot{\theta}_1$ 和 $\dot{\theta}_2$, 求解过程有复杂的符号代换, 可使用 matlab 进行符号求解, 使用如下代码:

```

1 % 符号定义
2 syms p_1 p_2 dtheta_1 dtheta_2 m1 m2 l1 l2 theta_1 theta_2 I1 I2
   dx1 dx2 dy1 dy2 L g y1 y2 dp1 dp2
3 % 物理量定义
4 I1=(1/12)*m1*l1^2;
5 I2 = (1/12)*m2*l2^2;
6 y1 = -(1/2)*l1*cos(theta_1);
7 dy1 = (1/2)*l1*sin(theta_1)*dtheta_1;
8
9 y2 = -l1*cos(theta_1)-(1/2)*l2*cos(theta_2);
10 dy2 = l1*sin(theta_1)*dtheta_1 + (1/2)*l2*sin(theta_2)*dtheta_2;
11
12 dx1 = (1/2)*l1*cos(theta_1)*dtheta_1;
13 dx2 = l1*cos(theta_1)*dtheta_1 + (1/2)*l2*cos(theta_2)*dtheta_2;
14
15 L = (1/2)*m1*(dx1^2+dy1^2) +(1/2)*m2*(dx2^2+dy2^2) +(1/2)*I1*(
   dtheta_1)^2 +(1/2)*I2*(dtheta_2)^2 -m1*g*y1-m1*g*y2;
16 lL = latex(L);
17 % 方程组求解
18 [soldtheta_1, soldtheta_2] = solve([p_1==diff(L,'dtheta_1'),p_2 =
   diff(L,'dtheta_2')],[dtheta_1,dtheta_2]);

```


经过求解得到：

$$f = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} \frac{12(2l_2 p_1 - 3l_1 p_2 \cos(\theta_1 - \theta_2))}{l_1^2 l_2 (8m_1 + 15m_2 - 9m_2 \cos(2\theta_1 - 2\theta_2))} \\ \frac{12(2l_1 m_1 p_2 + 6l_1 m_2 p_2 - 3l_2 m_2 p_1 \cos(\theta_1 - \theta_2))}{l_1 l_2^2 m_2 (8m_1 + 15m_2 - 9m_2 \cos(2\theta_1 - 2\theta_2))} \\ -\frac{l_1(3gm_1 \sin(\theta_1) + \theta_1 \dot{\theta}_2 l_2 m_2 \sin(\theta_1 - \theta_2))}{2} \\ -\frac{l_2(gm_1 \sin(\theta_2) - \dot{\theta}_1 \dot{\theta}_2 l_1 m_2 \sin(\theta_1 - \theta_2))}{2} \end{bmatrix}$$

3.2 Matlab 仿真

在 matlab 中使用四阶龙格库塔方法求解微分方程，给定初始位置和物理参数，可以对混沌二级摆系统进行仿真，并记录完整的 $\theta_1, \theta_2, p_1, p_2$ 数据：

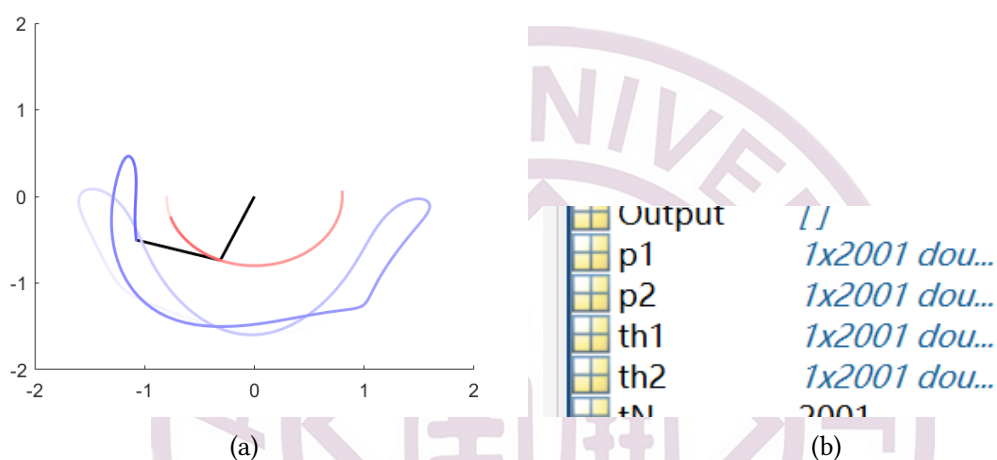


图 3-1

```

1 N=2;%双摆
2 m1=1.6;
3 m2=1.6;
4 l1=0.8;
5 l2=0.8;
6 g=9.8;
7 Input=[N,m1,m2,l1,l2,g];
8 %初始条件和时间设置
9 y0=[pi/2;pi/2;0;0];%这里全部是弧度值。分别代表[摆1与垂面夹角, 摆2与垂面夹角, 摆1角动量, 摆2角动量]

```

可以在代码中调整 m_1, m_2, l_1, l_2 进行仿真，得到输入输出数据组用于后续参数辨识。

第4章 麦夸特方法参数辨识

4.1 简单模型尝试

为了初步验证麦夸特方法的可行性，可先对简单非线性模型做尝试。

选择对简单指数函数做参数辨识，使用 matlab 编程实现

```

1 % 数据量
2 num = 10;
3 % 实际数据
4 x=[5 10 20 30 40 50 60 70 80 90];
5 y=[1.1 1.2 1.4 1.7 2.2 3.2 6.6 22.5 49.5 81.3];
6 % 初始化目标函数
7 J_aim = zeros(1,1000);

```

选用 10 组采样数据, 根据数据走势规律, 认为其是参数待定的指数函数, 初始化一个行向量用来存储每次迭代得到的目标函数值

```

1 % 计算第一个目标函数J_aim(1)
2 for j = 1 : num
3     J_aim(1)=J_aim(1)+(y(j)-parameter(1)^(x(j)-1))^2;
4 end

```

使用初始给定参数计算第一个目标函数值

```

1 while true
2     % 计算A和Q
3     A = 0;
4     Q = 0;
5     for i = 1:num
6         A = A + ((x(i)-1)*((parameter(t-1))^(x(i)-2)))^2;
7         Q = Q + (x(i)-1)*(parameter(t-1)^(x(i)-2))*(y(i)-parameter(t-1)^(x(i)-1));
8     end
9     while true
10        % 更新参数
11        v=-(A+(lambda/10)*1)^(-1)*(-Q);
12        parameter(t) = parameter(t-1) + v ;
13        % 计算在当前参数下的目标函数
14        for j = 1 : num
15            % 根据当前参数计算目标函数值
16            J_aim(t)=J_aim(t)+(y(j)-(parameter(t))^(x(j)-1))^2;
17        end

```

```

18         if J_aim(t) ≥ J_aim(t-1)
19             lambda = 10*lambda;
20             J_aim(t) = 0;
21         else
22             break;
23         end
24     end
25     a = abs(parameter(t)-parameter(t-1));
26     if a < e
27         break;
28     else
29         t=t+1;
30     end
31 end

```

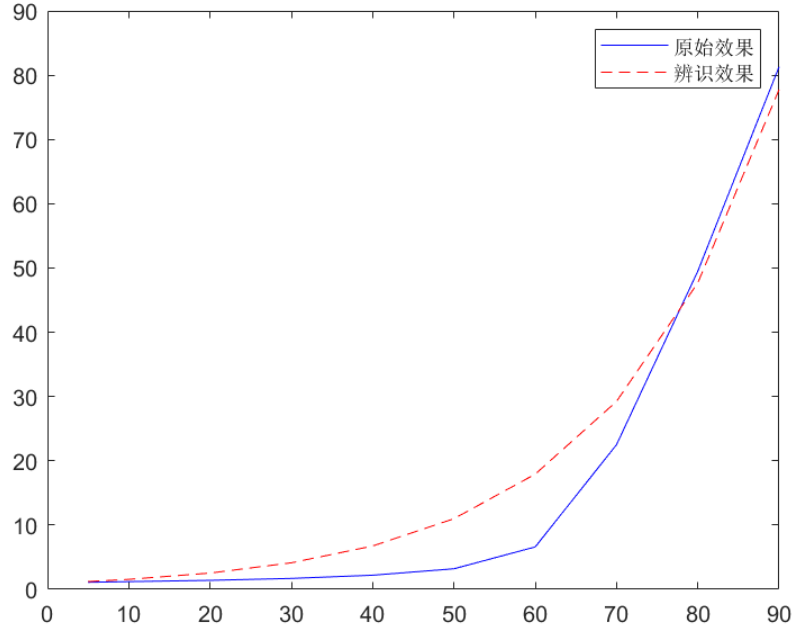
循环实现麦夸特算法。

在不同的给定初值下，能得到如下结果：

给定初值	辨识结果	迭代次数	目标函数最终值
1	1.0501	10	2.780064645662867e+02
1.5	1.0501	39	2.780064645662868e+02
3	1.0501	100	2.780064645662868e+02
5	1.0501	145	2.780064645662868e+02
50	1.0501	349	2.780064645662868e+02

可以看到麦夸特能够有不错的收敛辨识效果，即便给定的初始参数相差较大，仍能收敛在相同的位置，考虑可能是因为给定的指数结构较简单，参数个数仅一个，易收敛到全局最优处。

另外由于给定数据并非真实的指数函数采样，只是近似当作指数函数处理，所以目标函数仍数值较大，如下是给定数据和参数辨识结果的对比：



4.2 混沌二级摆辨识

由上章节已经得到了混沌二级摆的动力学方程：

$$f = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} \frac{12(2l_2 p_1 - 3l_1 p_2 \cos(\theta_1 - \theta_2))}{l_1^2 l_2 (8m_1 + 15m_2 - 9m_2 \cos(2\theta_1 - 2\theta_2))} \\ \frac{12(2l_1 m_1 p_2 + 6l_1 m_2 p_2 - 3l_2 m_2 p_1 \cos(\theta_1 - \theta_2))}{l_1 l_2^2 m_2 (8m_1 + 15m_2 - 9m_2 \cos(2\theta_1 - 2\theta_2))} \\ -\frac{l_1 (3g m_1 \sin(\theta_1) + \dot{\theta}_1 \dot{\theta}_2 l_2 m_2 \sin(\theta_1 - \theta_2))}{2} \\ -\frac{l_2 (g m_1 \sin(\theta_2) - \dot{\theta}_1 \dot{\theta}_2 l_1 m_2 \sin(\theta_1 - \theta_2))}{2} \end{bmatrix}$$

现考虑应用由仿真得到的输入数据： θ_1 θ_2 p_1 p_2

以及对输入数据做差分运算得到的导数值作为输出数据： $\dot{\theta}_1$ $\dot{\theta}_2$

以此来模拟实际现实中，当拿到一组二级摆的角度数据和角动量数据时如何辨识两杆的质量和杆长 m_1, m_2, l_1, l_2 。

在前述的 matlab 仿真中,已经得到了格式为 1×1000 的输入输出数据,包括 θ_1 , θ_2 , p_1 , p_2 , 使用 matlab 中求差分运算 `diff()` 得到 $\dot{\theta}_1$, $\dot{\theta}_2$ 现确定根据方程：

$$\dot{\theta}_1 = \frac{12(2l_2 p_1 - 3l_1 p_2 \cos(\theta_1 - \theta_2))}{l_1^2 l_2 (8m_1 + 15m_2 - 9m_2 \cos(2\theta_1 - 2\theta_2))}$$

进行参数辨识。

在这里一共有四个参数需要初始化：

```
1 % 给定第一个参数集
2 parameter = zeros(4,N);
```

```

3 parameter(1,1) = 1;
4 parameter(2,1) = 1;
5 parameter(3,1) = 1;
6 parameter(4,1) = 1;

```

首先在给定参数组下计算第一个目标函数 J_{aim0}

```

1 % 计算第一个目标函数J_aim0
2 for j = 1 : num
3     % 参数和数据传入
4     m1 = parameter(1,1); m2 = parameter(2,1); l1 = parameter(3,1);
5     l2 = parameter(4,1);
6     theta_1 = x(1,j); theta_2 = x(2,j); p_1 = x(3,j); p_2 = x(4,j);
7     dtheta_1(j) = (12*(2*l2*p_1 - 3*l1*p_2*cos(theta_1 - theta_2)))/(
8         l1^2*l2*(8*m1 + 15*m2 - 9*m2*cos(2*theta_1 - 2*theta_2)));
9
10    % 计算在当前参数下实际输出和辨识输出差
11    detlay(j) = y(j)-dtheta_1(j);
12    % 根据当前参数计算目标函数值
13    J_aim(1) = J_aim(1) + detlay(j)^2;
14 end

```

公式部分较复杂，可以使用 matlab 的符号求解功能辅助计算。

由于具有四个参数，在计算 A_i, Q_i 时，需要用到原函数对四个参数的一阶偏导数：

```

1 % 雅可比矩阵,用于计算预测函数的一阶微分,大小4*1
2 ddtheta1dm1 = -(96*(4*l2*p_1 - 6*l1*p_2*cos(theta_1 - theta_2)))/(l1
3     ^2*l2*(480*m1*m2 - 540*m2^2*cos(2*theta_1 - 2*theta_2) + 81*m2
4     ^2*cos(4*theta_1 - 4*theta_2) + 128*m1^2 + 531*m2^2 - 288*m1*m2*
5     cos(2*theta_1 - 2*theta_2)));
6 ddtheta1dm2 = -(48*(15*l2*p_1 - (63*l1*p_2*cos(theta_1 - theta_2))/4
7     - 9*l2*p_1*cos(2*theta_1 - 2*theta_2) + (27*l1*p_2*cos(3*
8     theta_1 - 3*theta_2))/4))/(l1^2*l2*(480*m1*m2 - 540*m2^2*cos(2*
9     theta_1 - 2*theta_2) + 81*m2^2*cos(4*theta_1 - 4*theta_2) + 128*
10    m1^2 + 531*m2^2 - 288*m1*m2*cos(2*theta_1 - 2*theta_2)));
11 ddtheta1dl1 = -(48*l2*p_1 - 36*l1*p_2*cos(theta_1 - theta_2))/(l1^3*
12    l2*(8*m1 + 15*m2 - 9*m2*cos(2*theta_1 - 2*theta_2)));
13 ddtheta1dl2 = (36*p_2*cos(theta_1 - theta_2))/(l1*l2^2*(8*m1 + 15*m2
14    - 9*m2*cos(2*theta_1 - 2*theta_2)));
15 J0=[ddtheta1dm1; ddtheta1dm2; ddtheta1dl1; ddtheta1dl2];
16 % 计算在当前参数下实际输出和辨识输出差
17 detlay(i) = y(i)-dtheta_1(i);
18 A = A + J0*J0';
19 Q = Q - J0 * detlay(i);

```

随后的循环过程按照正常麦夸特方法进行，最终将目标函数的优化过程记录在 J_{aim} 中，将参数优化过程记录在 $parameter$ 中。

实验中，首先将混沌二级摆仿真文件中的物理参数给定为 $m_1 = 1; m_2 = 1; l_1 = 1; l_2 = 1$ ，经过仿真运行得到数据组，再在麦夸特辨识文件中给定不同的参数初值进行辨识，效果如下：

给定初值	辨识结果	迭代次数	目标函数最终值
1, 1, 1, 1	1.0159, 0.9954, 1.0108, 1.0119	5	15.533692830758467
1.5, 1, 1, 1	1.2764, 1.2506, 0.9018, 0.9028	6	15.533692830758472
3, 1, 1, 1	2.2028, 2.1583, 0.6865, 0.6872	223	15.533692830761117
0.9, 1.1, 1, 1	1.0249, 1.0042, 1.0064, 1.0075	5	15.533692830761117
20, 20, 20, 20	13.1442, 12.8788, 0.2810, 0.2813	10807	15.533692830854473

从上述的几个初值尝试中，可以明显看到二级摆系统的辨识极易受初值给定影响，初值的不同引起的辨识结果天差地别，这套算法几乎不能提供有价值的辨识效果。

但同时读者考虑双摆动力学方程关于 θ 的其实有两个，即：

$$\dot{\theta}_1 = \frac{12 (2 l_2 p_1 - 3 l_1 p_2 \cos(\theta_1 - \theta_2))}{l_1^2 l_2 (8 m_1 + 15 m_2 - 9 m_2 \cos(2 \theta_1 - 2 \theta_2))}$$

$$\dot{\theta}_2 = \frac{12 (2 l_1 m_1 p_2 + 6 l_1 m_2 p_2 - 3 l_2 m_2 p_1 \cos(\theta_1 - \theta_2))}{l_1 l_2^2 m_2 (8 m_1 + 15 m_2 - 9 m_2 \cos(2 \theta_1 - 2 \theta_2))}$$

前述中，参数的辨识仅依赖第一个方程，但实际的参数需满足后面其他的动力学方程，遂考虑进一步将关于 $\dot{\theta}_2$ 的方程引入计算目标函数，具体实现需要修改目标函数计算过程已经 A_i, Q_i 更新过程：

```

1 % 计算第一个目标函数J_aim0
2 for j = 1 : num
3     % 参数和数据传入
4     m1 = parameter(1,1); m2 = parameter(2,1); l1 = parameter(3,1);
5     l2 = parameter(4,1);
6     theta_1 = x(1,j); theta_2 = x(2,j); p_1 = x(3,j); p_2 = x(4,j);
7     dtheta_1(j) = (12*(2*l2*p_1 - 3*l1*p_2*cos(theta_1 - theta_2)))/(
8         l1^2*l2*(8*m1 + 15*m2 - 9*m2*cos(2*theta_1 - 2*theta_2)));
9     dtheta_2(j) = (12*(2*l1*m1*p_2 + 6*l1*m2*p_2 - 3*l2*m2*p_1*cos(
10         theta_1 - theta_2)))/(l1*l2^2*m2*(8*m1 + 15*m2 - 9*m2*cos(2*
11         theta_1 - 2*theta_2)));
12     compute_output(:,j) = [dtheta_1(j);dtheta_2(j)];

```



```

9
10 % 计算在当前参数下实际输出和辨识输出差
11 detlay(:,j) = y(:,j)-compute_output(:,j);
12 % 根据当前参数计算目标函数值
13 J_aim(1) = J_aim(1) +sum(power(detlay(:,j),2));
14 end

```

其余代码部分不再过多展示，最终辨识效果如下：

仍将混沌二级摆仿真文件中的物理参数给定为 $m_1 = 1; m_2 = 1; l_1 = 1; l_2 = 1;$ ，经过仿真运行得到数据组，再在麦夸特辨识文件中给定不同的参数初值进行辨识：

给定初值	辨识结果	迭代次数	目标函数最终值
1, 1, 1, 1	1.0082, 1.0021, 1.0095, 1.0109	5	1.166443524375716e+02
1.5, 1, 1, 1	1.2633, 1.2557, 0.9018, 0.9031	6	1.166443524375716e+02
3, 1, 1, 1	2.2207, 2.2073, 0.6802, 0.6811	46	1.166443524375724e+02
0.9, 1.1, 1, 1	1.0168, 1.0107, 1.0052, 1.0066	6	1.166443524375713e+02
20, 20, 20, 20	-3.8204, 0.0004, -33.3874, -56.2591	11	5.305734590186911e+04

可以看到很遗憾效果几乎没有提升，唯一有进步的是收敛速度明显变快，但仍然无法收敛到真实参数处。

4.3 修正

经 2024.5.28 日与指导老师交流，意识到实现流程中有一严重纰漏，即上述的辨识结果中目标函数收敛值都出奇的大，理想的目标函数值应收敛到 0 附近，显然存在错误。

不难发现，在目标函数的计算中，“辨识数据”来自使用辨识参数带入理论公式进行计算，“实际数据”来自对 θ_1, θ_2 数据集取差分运算得到 $\dot{\theta}_1, \dot{\theta}_2$ ，问题即在于此：取差分运算模拟微分势必会带来误差，而这一误差在采样时间间隔大时十分严重，为了保证差分近似微分的准确性，应使采样间隔尽可能小，否则随着采样数据量的增多会大幅度增大目标函数因差分计算而引起的误差，使目标函数值不具有参考价值。

实践中将采样间隔缩小到之前的百分之一，相同仿真时间内数据量增加到原来的一百倍，重新进行辨识，发现无论初值如何给定，麦夸特算法均能快速迭代辨识出一组参数使得目标函数值基本收敛至零，但参数辨识结果仍不能与真实参数相符合，实际上这正是反映了混沌双摆系统显然存在多解问题，即便物理参数不同，仍可以出现

几乎相同的物理运动状态，故这类问题难以准确辨识到仿真使用的参数。



第 5 章 总结

本项目使用麦夸特算法，分别针对简单非线性系统（指数函数）和复杂强非线性系统（混沌二级摆）做尝试，最终从结果中可以看出：

- 麦夸特算法在简单非线性系统的参数辨识中有不错的效果，对于单参数指数系统的辨识较为准确；
- 在面对如混沌双摆系统的非线性系统时，考虑是由于系统目标函数局部最优点过多，麦夸特算法极易陷入局部最优的情况，对给定初值极为敏感，如若初值靠近真实参数，可在几步内得到良好辨识结果，但若初值过于随意，则大概率落在初值附近的局部最优解上跳出循环，导致辨识失败；

未来可进一步尝试单纯形算法，适当修改参数以克服强非详细系统参数辨识的局部最优问题。

本报告仅呈现关键部分及结论性内容，用于复现 or 参考的项目完整代码可查看 <https://github.com/Qiaozs/Mcquart-algorithm>