

统计推断在数模转换系统中的应用

第 054 组 刘海粟 5130309658 李斯达 5130309666

摘要: 本课程在以某电子产品内部传感传感器部件的输入输出特性为研究对象, 在确保测量精度的前提下, 运用统计方法结合 Matlab 四百多组样本中电压信号 X 和物理量 Y 的数据关系进行分析, 对样本数据进行拟合, 运用遗传算法为该模块的批量生产设计一种成本合理的传感特性校准(定标工序)方案。

关键词: 统计推断, 传感特性, 拟合, 遗传算法, 交叉变异, 拟合, 成本, Matlab

Application of Statistical Inference in AD&DA Inverting System

Group Number:054 Haisu Liu 5130309658 Sida Li 5130309666

ABSTRACT: This course in order to input and output characteristics of the internal sensor components of some electronic products as the research object, on the premise of ensuring accuracy, were analyzed by using statistical method in combination with Matlab more than 400 samples the voltage signals X and Y data between physical quantities, by fitting the sample data, and use the genetic algorithm to design a sensor characteristics calibration the reasonable cost for the module of batch production (calibration procedure) scheme.

Key words: Statistic Interference, Sensing characteristics, Fit, Genetic Algorithm, crossover and mutation, Cost, Matlab

1. 引言

假定有某型投入批量试生产的电子产品, 其内部有一个模块, 功能是监测某项与外部环境有关的物理量(可能是温度、压力、光强等)。该监测模块中传感器部件的输入输出特性呈明显的非线性。检测的工序一方面要保证精度, 另一方面要考虑到控制操作成本, 因此我们需要研究该系统的特性来找出一种方法衡量产品的品质。

本次研究中, 我们运用遗传算法, 结合拟合方法对近五百组数据进行分析, 结合 Matlab 语言, 找出成本相对最小的传感特性校准(定标工序)方案。

2. 对系统简单描述

为了对本课题展开有效讨论, 需建立一个数学模型, 对问题的某些方面进行必要的描述和限定。所谓传感特性校准, 就是针对某一特定传感部件个体, 通过有限次测定, 估计其 Y 值与 X 值间一一对应的特性关系的过程。数学上可认为是确定适用于该个体的估测函数 $\hat{y} = f(x)$ 的过程, 其中 x 是 X 的取值, \hat{y} 是对应 Y 的估测值。

考虑实际工程中该监测模块的应用需求, 同时为便于在本课题中开展讨论, 我们将问题限于 X 为离散取值的情况, 规定

$$X \in \{x_1, x_2, x_3, \dots, x_{50}, x_{51}\} = \{5.0, 5.1, 5.2, \dots, 9.9, 10.0\}$$

相应的 Y 估测值记为 $\hat{y}_i = f(x_i)$, Y 实测值记为 y_i , $i = 1, 2, 3, \dots, 50, 51$ 。

评价函数:

[A]单点定标误差成本

$$s_{i,j} = \begin{cases} 0 & \text{if } |\hat{y}_{i,j} - y_{i,j}| \leq 1 \\ 1 & \text{if } 1 < |\hat{y}_{i,j} - y_{i,j}| \leq 2 \\ 2 & \text{if } 2 < |\hat{y}_{i,j} - y_{i,j}| \leq 3 \\ 4 & \text{if } 3 < |\hat{y}_{i,j} - y_{i,j}| \leq 5 \\ 10 & \text{if } |\hat{y}_{i,j} - y_{i,j}| > 5 \end{cases} \quad (1)$$

单点定标误差的成本按式 (1) 计算, 其中 $y_{i,j}$ 表示第 i 个样本之第 j 点 Y 的实测值, $\hat{y}_{i,j}$

表示定标后得到的估测值 (读数), 该点的相应误差成本以符号 $s_{i,j}$ 记。

[B]单点测定成本

实施一次单点测定的成本以符号 q 记。本课题指定 $q=12$ 。

[C] 某一样本个体的定标成本

$$S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i \quad (2)$$

对样本 i 总的定标成本按式 (2) 计算, 式中 n_i 表示对该样本个体定标过程中的单点测定次数。

[D] 校准方案总体成本

按式 (3) 计算评估校准方案的总体成本, 即使用该校准方案对标准样本库中每个样本个体逐一定标, 取所有样本个体的定标成本的统计平均。

$$C = \frac{1}{M} \sum_{i=1}^M S_i \quad (3)$$

3. 算法设计实现与性能分析

3.1 遗传算法的基本原理

遗传算法 (Genetic Algorithm, 简称 GA) 是模拟生物进化过程的计算模型, 是自然遗传学与计算机科学相互结合、相互渗透而形成的新的计算方法。遗传是一种生物从其亲代继承特性和性状的现象。继承的信息由基因携带, 多个基因组成染色体, 基因在染色体中的位置为基因座 (Locus)。同一基因座可能有的全部基因为等位基因 (Alleles, 等位基因和基因座决定了染色体的特征, 也决定了生物个体的特性。从染色体的表现形式看, 有两种相应的表示模式, 分别为基因型 (Genotype) 和表现型 (Phenotype)。表现型是指生物个体表现出来的性状, 而基因型则是指与表现密切相关的基因组成。同一基因型的生物个体在不同的环境条件下有不同的表现型。因此, 表现型是基因型与环境相互作用。

在遗传算法中染色体对应的是一系列符号序列, 在标准的遗传算法 (即基本遗传算法) 中, 通常用 0, 1 组成的位串表示, 串上各个位置对应基因座, 各位置上的取值对应等位基因。遗传算法对染色体进行处理, 染色体称为基因个体。一定数量的基因个体组成基因种群。种群中个体的数目为种群的规模, 各个体对环境的适应程度称为适应度 (Fitness)。适应度

越大，说明样本越好。本实验就是运用遗传算法中的遗传、交叉、变异等找出适应度最高的样本。

3.2 算法描述

[A]对于这个问题，如果要测量所有的测试点必定要耗费大量的成本，若是选取其中几个点，再取遍所有情况（以 7 点为例，就要有 115775100 种情况），很明显由于数据庞大，要比较全部的方案，成本太高，穷举法不可取。

但可以考虑先使用穷举法进行优秀样本的选择，用于充当初始父代。

[B]我们采用老师课堂上提到的遗传算法来解决这个问题，下面是遗传算法的步骤：

1、初始化：设置进化代数计数器 $t=0$ ，设置最大进化代数 50 代，随机生成 51 个初始染色体

2、个体评价：对选择方式进行打分

3、交叉运算：根据交叉概率判断是否交叉（交叉概率为 0.8）

4、变异运算：根据变异概率判断每一位点是否变异（变异概率为 0.2）

5、修复运算：对不合法的子代进行修复（按照多减少补并且随机的原则）

6、终止条件判断:若成本小于90时，则以进化过程中所得到的具有最大适应度个体作为最优解输出，程序终止。

3.3 拟合方法

多项式拟合

实验所提供的数据曲线如下图所示绝大部分都很接近一个多项式曲线。根据初稿的对实验数据的拟合，多项式拟合出的多项式曲线更具有拟合的准确性。在本实验中采用最高次项为三次的多项式拟合。

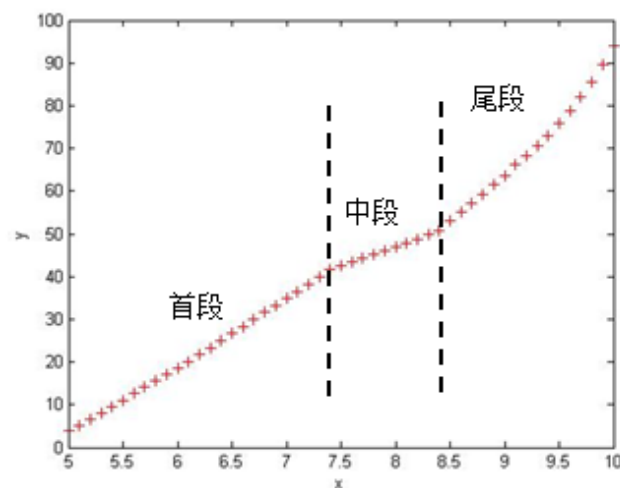
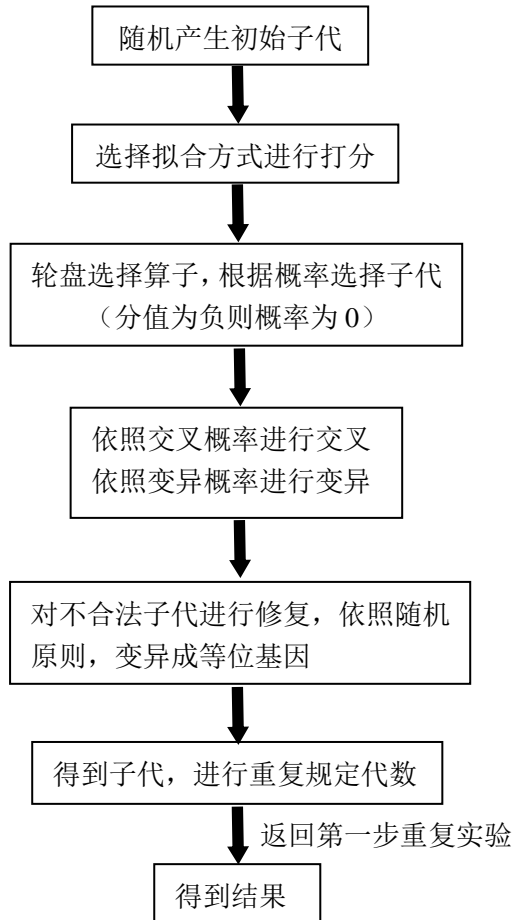


图 1 传感特性图示

3.4 遗传算法的实现



我们采用了一个有 100 个种群，每个种群的染色体为 51 条，子代个数为 20 代，交叉概率为 1.0，变异概率为 0.8 的遗传算法。因为之前我们是采用随机从 51 个点选取出来几个点再进行打分、比较、选择、交叉、变异，而且 $51 \times 51 \ll 100^{20}$ ，所以我觉得我们的方法几乎已经遍历了所有的可能选点方案，所以出来的结果是符合要求的。

程序具体见附页。

3.5 结果分析

由 MATLAB 的运行结果，使用 Origin 作图，可得，

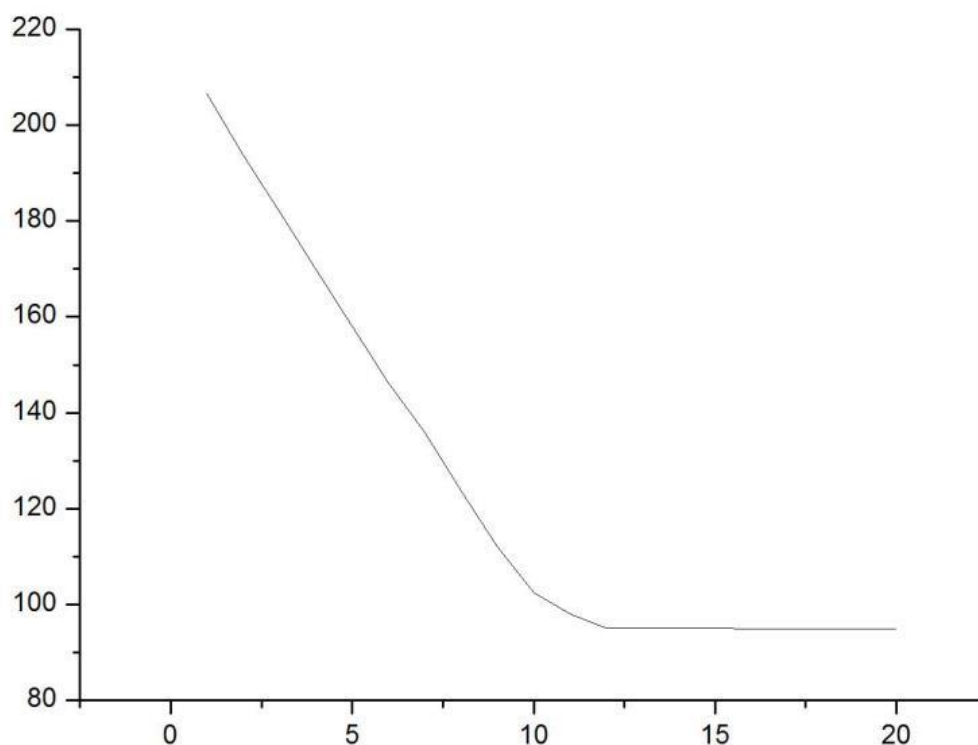


图 2 最优成本收敛曲线

图中横轴代表为计算代数，纵轴为每一代的最优成本

最佳取点方案 $q = 4, 15, 25, 33, 43, 50$

最优取样成本为 89.66

4. 结论

1、通过使用遗传算法，我们得到：4、15、25、33、43、50 这 7 个数据点后可采用三次多项式拟合法估计整条曲线。

2、保留最大值的遗传算法能加快遗传算法的收敛性，同时保证最优解不重新打破。

3、模拟生物的多种算法可以运用于工程实践中来。

5. 收获

在本学期的统计推断课程中，我们小组努力从对课程的分析、方案的讨论、编程实现、到最后方案评估，我们和其他小组的同学相互学习，相互进步，那种团结合作以及最后的默契配合，让我们每一个成员都非常珍惜和怀念。不仅如此，我们通过本课程，初步学会了数学建模，以及方案优劣的综合分析，以及应用 MATLAB 进行基本的程序编写，这个软件不仅为我们分析过程中提供了很大的方便，使我们学到了很多软件编写方面的技巧，而且主要是为统计推断课程做出了不少贡献，为课程以后的开展作了一定程度的贡献。

最后感谢老师和助教在平时学习中对我们遇到的问题加以指导，使我们成功地完成了此次任务！

6. 参考文献

[1] 张德丰. MATLAB 数字信号处理[M]. 清华大学出版社，2010:01.

[2] 《遗传算法和组合优化》[M].

- [3] 上海交通大学数学系. 《概率论与数理统计》[M]. 上海交通大学出版社, 2011:05.
- [4] 《DS 证据理论在雷达体制识别中的应用》(王 勇, 毕大平 文献标识码: A 文章编号: CN51—1418(2005)06—0003—04)

附录:

代码清单 1: (主体函数)

```
nvars=6;
LB=[1 1 1 1 1 1];
UB=[51 51 51 51 51 51];

options=gaoptimset('MutationFcn',@mutationadaptfeasible, ...
    'Display','iter');
fitnessfcn=@(x) custom_fitness(x);
[x,fval]=ga(fitnessfcn,nvars,[],[],[],[],LB,UB,[],1:6,options);
```

代码清单 2: (成本函数)

```
function value=custom_fitness(x)
value=0;
dataform =
importdata('C:\Users\lenovo\Documents\MATLAB\dataform.csv');
for i=1:469

y=dataform(2*i,x);
fitobject=fit(dataform(1,x)',y','smoothingspline');
sum=0;
for j=1:51
dif=abs(dataform(2*i,j)-fitobject(dataform(1,j)));
if dif<=0.5
    sum=sum+0;
elseif dif<=1
    sum=sum+0.5;
elseif dif<=2
    sum=sum+1.5;
elseif dif<=3
    sum=sum+6;
elseif dif<=5
    sum=sum+12;
else sum=sum+25;
end
```

```

end
value=value+sum+72;
end
value=value/469;
end

```

代码清单 3: (ga 函数)

```

function [x,fval,exitFlag,output,population,scores] =
ga(fun,nvars,Aineq,bineq,Aeq,beq,lb,ub,nonlcon,intcon,options)
defaultopt = struct('PopulationType', 'doubleVector', ...
    'PopInitRange', [], ...
    'PopulationSize', '50 when numberOfVariables <= 5, else 200', ...
    'EliteCount', '0.05*PopulationSize', ...
    'CrossoverFraction', 0.8, ...
    'MigrationDirection','forward', ...
    'MigrationInterval',20, ...
    'MigrationFraction',0.2, ...
    'Generations', '100*numberOfVariables', ...
    'TimeLimit', inf, ...
    'FitnessLimit', -inf, ...
    'StallTest', 'averageChange', ...
    'StallGenLimit', 50, ...
    'StallTimeLimit', inf, ...
    'TolFun', 1e-6, ...
    'TolCon', 1e-3, ...
    'InitialPopulation',[], ...
    'InitialScores', [], ...
    'NonlinConAlgorithm', 'auglag', ...
    'InitialPenalty', 10, ...
    'PenaltyFactor', 100, ...
    'PlotInterval',1, ...
    'CreationFcn',@gacreationuniform, ...
    'FitnessScalingFcn', @fitscalingrank, ...
    'SelectionFcn', @selectionstochunif, ...
    'CrossoverFcn',@crossoverscattered, ...
    'MutationFcn',{@mutationgaussian 1 1}}, ...
    'HybridFcn',[], ...
    'Display', 'final', ...
    'PlotFcns', [], ...
    'OutputFcns', [], ...
    'Vectorized','off', ...
    'UseParallel', false);

try

```

```

    narginchk(1,11);
catch ME
    error(message('globaloptim:ga:numberOfInputs', ME.message));
end

if nargin == 1 && nargout <= 1 && isequal(fun,'defaults')
    x = defaultopt;
    return
end

if nargin < 11, options = [];
    if nargin < 10, intcon = [];
        if nargin < 9, nonlcon = [];
            if nargin < 8, ub = [];
                if nargin < 7, lb = [];
                    if nargin < 6, beq = [];
                        if nargin < 5, Aeq = [];
                            if nargin < 4, bineq = [];
                                if nargin < 3, Aineq = [];
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

if nargin == 3 && isstruct(Aineq)
    options = Aineq; Aineq = [];
end

if nargin == 10 && isstruct(intcon)
    options = intcon;
    intcon = [];
end

if nargin == 1
    if isa(fun,'struct')

[fun,nvars,Aineq,bineq,Aeq,beq,lb,ub,nonlcon,intcon,rngstate,options]
= separateOptimStruct(fun);
        resetDfltRng(rngstate);
    end
end

```



```

        else
            error(message('globaloptim:ga:invalidStructInput'));
        end
    end
end

if iscell(fun)
    FitnessFcn = fun{1};
else
    FitnessFcn = fun;
end

if isempty(FitnessFcn) || ~(isa(FitnessFcn,'inline') ||
isa(FitnessFcn,'function_handle'))
    error(message('globaloptim:ga:needFunctionHandle'));
end

valid = isnumeric(nvars) && isscalar(nvars)&& (nvars > 0) ...
    && (nvars == floor(nvars));
if ~valid
    error(message('globaloptim:ga:notValidNvars'));
end

defaultopt.PopInitRange = [-10;10];

if ~isempty(intcon)
    gaminlpvalidateoptions(options);
    defaultopt.PopulationSize = max(min(10*nvars, 100), 40);
    defaultopt.EliteCount = ceil(0.05*defaultopt.PopulationSize);
    defaultopt.PopInitRange = [-1e4 + 1; 1e4 + 1];
end

user_options = options;
if ~isempty(options) && ~isa(options,'struct')
    error(message('globaloptim:ga:optionsNotAStruct'));
elseif isempty(options)
    options = defaultopt;
end

options = gaoptimset(defaultopt,options);

options.UserSpecPopInitRange = isa(user_options, 'struct') && ...
    isfield(user_options, 'PopInitRange') &&
    ~isempty(user_options.PopInitRange);

msg = isoptimargdbl('GA', {'NVARs','A', 'b',

```

```

'Aeq','beq','lb','ub'}, ...
                                nvars, Aineq, bineq, Aeq, beq, lb,
ub);
if ~isempty(msg)
    error('globaloptim:ga:dataType',msg);
end

[x,fval,exitFlag,output,population,scores,FitnessFcn,nvars,Aineq,bineq,Aeq,beq,lb,ub, ...
NonconFcn,options,Iterate,type] =
gacommon(nvars,fun,Aineq,bineq,Aeq,beq,lb,ub, ...

nonlcon,intcon,options,user_options);

if exitFlag < 0
    return;
end

if isempty(Aineq)
    Aineq = zeros(0,nvars);
end
if isempty(bineq)
    bineq = zeros(0,1);
end
if isempty(Aeq)
    Aeq = zeros(0,nvars);
end
if isempty(beq)
    beq = zeros(0,1);
end

if ~isempty(intcon)
    [x,fval,exitFlag,output,population,scores] =
gaminlp(FitnessFcn,nvars, ...

Aineq,bineq,Aeq,beq,lb,ub,NonconFcn,intcon,options,output,Iterate);
else
    switch (output.problemtype)
        case 'unconstrained'
            [x,fval,exitFlag,output,population,scores] =
gaunc(FitnessFcn,nvars, ...
options,output,Iterate);
        case {'boundconstraints','linearconstraints'}
            [x,fval,exitFlag,output,population,scores] =

```

```

galincon(FitnessFcn,nvars, ...
        Aineq,bineq,Aeq,beq,lb,ub,options,output,Iterate);
    case 'nonlinearconstr'
        if strcmpi(options.NonlinConAlgorithm,'auglag')
            [x,fval,exitFlag,output,population,scores] =
gacon(FitnessFcn,nvars, ...

Aineq,bineq,Aeq,beq,lb,ub,NonconFcn,options,output,Iterate,type);
        else
            [x,fval,exitFlag,output,population,scores] =
gapenalty(...
            FitnessFcn,nvars,Aineq,bineq,Aeq,beq,lb,ub, ...
            NonconFcn,[],options,output,Iterate,type);
        end
    end
end
end

```