

基于 PCA 算法的故障诊断步骤

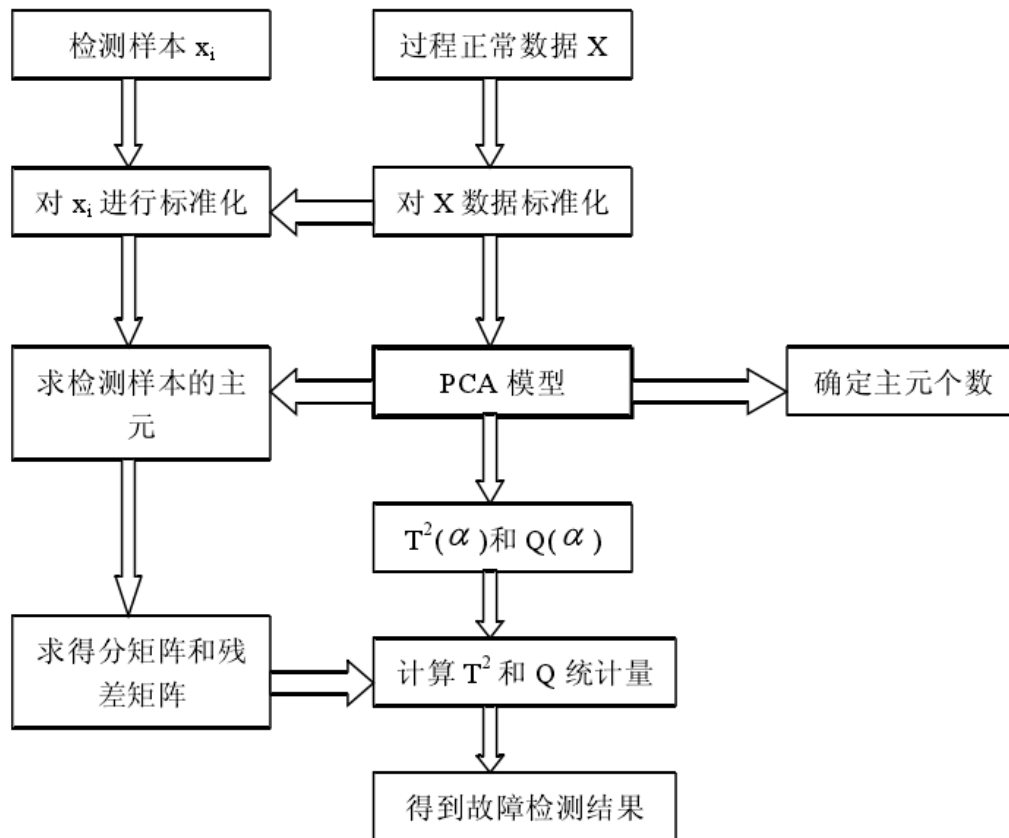


图 2-1 PCA 故障检测流程

离线 PCA 监测模型的计算步骤:

(1) 选择监控变量，收集正常工况下的各变量的样本，记为训

练样本数据 X_{train} 和检验数据 X_{test} ;

$$X_{\text{train}} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix}$$

X_{train} 为 $n \times m$ 矩阵，即 n 个样本， m 个观测变量（即以列向量来看的话，为一个观测变量各个采样点的值）

对样本数据 X_{train} 和检验数据 X_{test} 进行中心化和标准化处理

得到 $X_{s\text{train}}$ 和 $X_{s\text{test}}$;

中心化处理：按列对 X_{train} 减去观测变量的均值

$$x_{\cdot j} = x_{\cdot j} - \overline{x_{\cdot j}}$$

观测变量某一采样点的值减去这一观测变量所有采样点的平均值

$$\overline{x_{\cdot j}} = \frac{1}{n} \sum_{i=1}^n x_{ij}$$

求取一列（即某一观测变量）的平均值

标准化处理：对 X_{train} 除以观测变量的标准差（按列（观测变量）进行）

$$x_{s \cdot j} = \frac{x_{\cdot j}}{s_j}$$

$$s_j = \sqrt{\frac{\sum_{i=1}^n (x_{ij} - \overline{x_{\cdot j}})^2}{n-1}}$$

标准差

求出标准化矩阵 $X_{s\text{train}}$ 的协方差矩阵 Σ ;

$X_{s\text{train}}$ 的协方差矩阵 Σ 为:

$$\Sigma = \frac{1}{n-1} X_{s\text{train}}^T * X_{s\text{train}}$$

(2) 对 Σ 进行特征分解，求得特征值 $\lambda_1, \lambda_2, \dots, \lambda_m$

($\lambda_1 > \lambda_2 > \dots > \lambda_m$) 及其对应的特征向量 p_1, p_2, \dots, p_m

(负荷向量);

(3) 确定主元个数, 确定了主元个数 k , 就得到了 k 个特征值

$\lambda_1 > \lambda_2 > \dots > \lambda_k$, 及其对应的特征向量 p_1, p_2, \dots, p_k ;

A: 累计贡献率法:

前 k 个主元的累积方差贡献率为: $\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^m \lambda_i}$

当前 k 个主元的累积方差贡献率达到 85%, 则主元个数取 k 值

B: 交叉检验估计法:

将采集到的数据分成 k 个部分, 1 部分数据用来建立主元模型, 剩下的 $k-1$ 部分用来作为检验数据去检验所建的模型。如此, 建立若干个不同主元个数的模型, 并测试所建立的模型, 从中选取一个通过检验后误差最小的模型的主元个数作为系统主元个数。

(4) 建立 PCA 主元模型, 并进行交叉验证以确定误差最小

按照 $t_i = X_{s\text{train}} P_i$, 求出第 i 个主元, 并依据

$$\begin{aligned}\widehat{X_{s\text{train}}} &= t_1 p_1^T + t_2 p_2^T + \dots + t_k p_k^T \\ &= X_{s\text{train}} p_1 p_1^T + X_{s\text{train}} p_2 p_2^T + \dots + X_{s\text{train}} p_k p_k^T\end{aligned}$$

求出其主元模型

用 $X_{s\text{test}}$ 带入得到另一主元模型 $\widehat{X_{s\text{test}}}$, 依据 $E = X - \hat{X}$, 求出模型

误差, 确定模型误差最小的那个模型即为主元模型。

(5) 计算 T^2 统计量控制限和 SPE 统计量控制限;

对于样本个数为 n , 主元个数为 k 的过程变量 X_{train} , T^2 统计量

服从自由度为 k 和 $n - k$ 的 F 分布, 则置信度为 α 的 T^2 统计量控制上限为:

$$T^2_{UCL} = \frac{k(n-1)}{n-k} \cdot F_{\alpha}(k, n-k)$$

$$\text{或 } T^2_{UCL} = \frac{k(n-1)(n+1)}{n(n-k)} \cdot F_{\alpha}(k, n-k)$$

检验水平为 α 的 SPE 统计量控制上限为:

$$Q_{UCL} = \theta_1 \left[1 + \frac{h_0 C_{\alpha} \sqrt{2\theta_2}}{\theta_1} + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0}$$

$$\theta_1 = \sum_{i=k+1}^m \lambda_i, \theta_2 = \sum_{i=k+1}^m \lambda_i^2, \theta_3 = \sum_{i=k+1}^m \lambda_i^3, h_0 = \frac{1-2\theta_2\theta_3}{3\theta_2^2},$$

C_{α} 是与 $(1-\alpha)$ 分位点对应的标准差

在线过程监测与故障诊断步骤:

(1) 采集第 i 时刻的在线实时数据 $\mathbf{X}_{(i)}$ ($\mathbf{X}_{(i)}$ 为 $1 \times m$ 矩阵), 并

进行中心化和标准化处理得到 $\mathbf{X}_{(i)_s}$;

(2) 按照 $\mathbf{t}_{(i)}(1 \times k) = \mathbf{X}_{(i)_s}(1 \times m)\mathbf{p}_k(m \times k)$, 求出 $\mathbf{X}_{(i)_s}$ 的得分

向量, 依据 $\widehat{\mathbf{X}}_{(i)_s} = \mathbf{t}_i \mathbf{p}_k^T(k \times m)$, 求出 PCA 模型估计量

$\widehat{\mathbf{X}}_{(i)_s}$, 这里 $\mathbf{p}_k = [p_{1k}, p_{2k}, \dots, p_{kk}]$;

(3) 计算 $\mathbf{X}_{(i)_s}$ 的 T^2 统计量和 SPE 统计量, 并画出 T^2 统计量和

SPE 统计量的控制图;

$$T_{(i)}^2 = t_{(i)} \Lambda_k^{-1} t_{(i)}^T = X_{(i)s} p_k \Lambda_k^{-1} p_k^T X_{(i)s}^T$$

$$Q_{(i)} = X_{(i)s} (I - p_k p_k^T) X_{(i)s}^T$$

(4) 将上述计算结果与 T^2 统计量控制限和 SPE 统计量控制限比较, 以检测过程运行有无异常, 当有异常状态发生时, 绘制贡献图, 找出与故障相关的系统变量:

1) 检查每个观测值 x 的标准化得分 $\frac{t_i}{\lambda_i}$, 并确定造成失控

状态的 r ($r < a$) 个得分: $\frac{t_i}{\lambda_i} < \frac{T^2_{UCL}}{a}$;

2) 计算每个变量 x_i 相对于失控得分 t_i 的贡献率是:

$$cont_{i,j} = \frac{t_i}{\lambda_i} p_{i,j} x_j$$

3) 当 $cont_{i,j}$ 是负时, 设它为零;

4) 计算第 j 个过程变量 x_j 的总贡献率:

$$CONT_j = \sum_{i=1}^r (cont_{i,j})$$

5) 把所有 m 个过程变量 x_j 的 $CONT_j$ 画在一个曲线图上。

PCA_TE 仿真程序:

%TE 过程的传统主元分析在 Matlab 中的仿真程序

%建立模型:

%载入模型数据, 以故障 11 为例

```
Xtrain = load('G:d11.dat');
```

```
Xtrain = double(Xtrain);
```

```
%载入测试数据
```

```
Xtest = load('G:d11_te.dat');
```

```
Xtest = double(Xtest);
```

```
%标准化处理:
```

```
X_mean = mean(Xtrain); %按列求 Xtrain 平均值
```

```
X_std = std(Xtrain); %求标准差
```

```
[X_row,X_col] = size(Xtrain); %求 Xtrain 行、列数
```

```
% for i = 1:X_col
```

```
%Xtrain(:,i)=(Xtrain(:,i) - X_mean(i)./X_std(i));
```

```
%Xtest(:,i) = (Xtest(:,i) - X_mean(i)./X_std(i));
```

```
% end
```

```
Xtrain=(Xtrain-repmat(X_mean,X_row,1))./repmat(X  
_std,X_row,1);
```

```
%求协方差矩阵
```

```
sigmaXtrain = cov(Xtrain);
```

```
%对协方差矩阵进行特征分解, lamda 为特征值构成的对
```

角阵，T 的列为单位特征向量，且与 lamda 中的特征值一一对应：

```
[T, lamda] = eig(sigmaXtrain);  
% disp('特征根（由小到大）');  
% disp(lamda);  
% disp('特征向量：');  
% disp(T);
```

%取对角元素(结果为一列向量)，即 lamda 值，并上下反转使其**从大到小**排列，主元个数初值为 1，若累计贡献率小于 90%则增加主元个数

```
D = flipud(diag(lamda));  
num_pc = 1;  
while sum(D(1:num_pc))/sum(D) < 0.9  
    num_pc = num_pc +1;  
end
```

%取与 lamda 相对应的特征向量

```
P = T(:, X_col-num_pc+1:X_col);
```

%求置信度为 99%、95%时的 T2 统计控制限

```
T2UCL1=num_pc*(X_row-1)*(X_row+1)*finv(0.99, num_
```

```
pc, X_row - num_pc) / (X_row * (X_row - num_pc));
T2UCL2 = num_pc * (X_row - 1) * (X_row + 1) * finv(0.95, num_
pc, X_row - num_pc) / (X_row * (X_row - num_pc));
```

%置信度为 99%的 Q 统计控制限

```
for i = 1:3
    theta(i) = sum((D(num_pc+1:X_col)).^i);
end
h0 = 1 - 2*theta(1)*theta(3)/(3*theta(2)^2);
ca = norminv(0.99, 0, 1);
QUCL = theta(1)*(h0*ca*sqrt(2*theta(2)))/theta(1)
+ 1 + theta(2)*h0*(h0 - 1)/theta(1)^2^(1/h0);
```

%在线监测:

%标准化处理

```
n = size(Xtest, 1);
Xtest = (Xtest - repmat(X_mean, n, 1)) ./ repmat(X_std, n
, 1);
```

%求 T2 统计量, Q 统计量

```
[r, y] = size(P*P');
I = eye(r, y);
```



```

T2 = zeros(n, 1);
Q = zeros(n, 1);
for i = 1:n
    T2(i)=Xtest(i,:)*P*inv(lamda(52-num_pc+1:52,
    52-num_pc+1:52))*P'*Xtest(i,:)' ;
    Q(i) = Xtest(i,:)*(I - P*P')*Xtest(i,:)' ;
end

```

%绘图

```

figure
subplot(2, 1, 1);
plot(1:n, T2, 'k');
title('主元分析统计量变化图');
xlabel('采样数');
ylabel('T^2');
hold on;

line([0, n], [T2UCL1, T2UCL1], 'LineStyle', '--', 'Color', 'r');

line([0, n], [T2UCL2, T2UCL2], 'LineStyle', '--', 'Col

```

```
or', 'g');
```

```
subplot(2, 1, 2);  
plot(1:n, Q, 'k');  
xlabel('采样数');  
ylabel('Q');  
hold on;
```

```
line([0, n], [QUCL, QUCL], 'LineStyle', '--', 'Color',  
'r');
```

%贡献图

%1. 确定造成失控状态的得分

```
S = Xtest(400, :)*P(:, 1:num_pc);  
r = [ ];  
for i = 1:num_pc  
    if S(i)^2/lamda(i) > T2UCL1/num_pc  
        r = cat(2, r, i);  
    end  
end
```

%2. 计算每个变量相对于上述失控得分的贡献

```

cont = zeros(length(r),52);
for i = length(r)
    for j = 1:52
        cont(i,j) =
abs(S(i)/D(i)*P(j,i)*Xtest(400,j));
    end
end

```

%3. 计算每个变量的总贡献

```

CONTJ = zeros(52,1);
for j = 1:52
    CONTJ(j) = sum(cont(:,j));
end

```

%4. 计算每个变量对 Q 的贡献

```

e = Xtest(400,:)*(I - P*P');
contq = e.^2;

```

%5. 绘制贡献图

```

figure;
subplot(2,1,1);
bar(CONTJ,'k');

```

```
xlabel('变量号');  
ylabel('T^2 贡献率 %');
```

```
subplot(2, 1, 2);  
bar(contq, 'k');  
xlabel('变量号');  
ylabel('Q 贡献率 %');
```