

摘要

本项目的核心任务是通过统计过程控制（SPC）方法，对某工厂生产的滚珠直径数据进行产品质量管理，评估产品工艺水平及其生产过程是否受控。统计过程控制作为一种基于概率统计的过程控制方法，自1924年由Shewhart博士提出控制图以来，已广泛应用于现代制造过程的质量控制中。本项目基于Matlab软件开发，对于收集到的数据，通过描述性统计分析、正态性检验、总体均值检验、工序能力指数计算与绘制均值控制图和方差控制图等多种方式，对数据进行全面分析，最终评估工艺水平及生产过程的受控状态。具体而言，首先，在数据收集过程中，结合实际数据与模拟数据，得到25组产品质量样本数据（每组至少5个样本）；随后，通过描述性统计分析方法对数据进行宏观的认知，根据均值、方差、极差、直方图等指标对数据进行初步分析；接着又利用假设检验的方法，分别通过Pearson卡方检验与t检验等方法验证数据是否符合正态分布，并对其总体均值进行检验。基于对于正态总体及其均值、方差的检验合理性，最终计算并评估了数据的工序能力指数，同时绘制了均值控制图和方差控制图，以直观反映生产过程中的数据波动情况。通过多种方法的综合分析，最终得出了工厂生产的滚珠直径的工艺水平评价和生产过程的受控状态判断。

关键词: 产品质量管理、统计过程控制（SPC）、假设检验、Shewhart控制图、Matlab

目录

1	项目概述	3
1.1	问题背景	3
1.2	项目任务	3
2	项目过程	3
2.1	数据收集	4
2.2	描述性统计分析	6
2.3	数据正态性检验	8
2.4	总体均值检验	10
2.5	数据工序能力指数的计算与评估	12
2.6	均值控制图与方差控制图	14
2.6.1	均值控制图	14
2.6.2	方差控制图	16
3	产品质量分析结论	18
	References	19

1 项目概述

1.1 问题背景

现代产品质量管理主要有两个核心目标：第一，评价产品的工艺水平如何，即产品质量如何；第二，评价产品的生产过程是否受控，即产品可靠性如何。从现代质量控制的角度看，一方面要求产品质量好，另一方面还要求可靠性高。

统计过程控制(Statistical Process Control, SPC)是一种基于概率统计的过程控制方法。1924年美国贝尔实验室的Shewhart博士提出控制图，标志着产品统计质量控制的正式起点。作为现代质量管理的重要方法，SPC广泛应用于现代制造过程的质量控制。20实际80年代，国际上半导体制造过程已经普遍采用SPC技术提升产品合格率和可靠性。比如Motorola公司提出并在美国通用电器公司等广泛应用的6 σ 管理，其主要技术基础就是SPC理论。

生产过程是否处于统计受控状态，取决于生产过程是否存在异常因素导致产品质量的起伏变化。产品加工结果是否满足加工规范要求，反应的是工艺水平的高低；而工艺是否受控，反应的是生产过程是否存在异常因素。

1.2 项目任务

在本次项目中，需要收集批量产品质量数据，利用SPC方法对其进行管理，分析整体工艺水平并判断生产过程是否处于统计受控状态，具体而言可细分为如下任务：

- 任务 1:收集或生成25组以上的数据，每组至少5个样本。
- 任务 2:根据均值，方差，极差，直方图等指标，对数据进行描述性统计分析。
- 任务 3:对数据进行正态性检验与总体的均值检验。
- 任务 4:计算数据工序能力指数并对其进行评估。
- 任务 5:描绘均值控制图和方差控制图。
- 任务 6:得出结论：工艺水平如何；生产过程是否处于统计受控状态。

2 项目过程

本项目完全基于Matlab代码实现SPC方法进行产品质量管理。

2.1 数据收集

本项目中的数据收集采用实际数据与模拟数据结合的方式，其中实际数据以书本7.4节的例题7.4.4^[1]为主要来源，在此基础上利用生成的模拟数据对其进行扩充，最终得到25组产品质量样本数据，每组中含有5个数据。

具体而言，本项目的问题情境为对工厂生产的滚珠直径（单位：*mm*）进行产品质量管理。书本上的例题提供了50个直径数据，将其作为扩充生成模拟数据的源数据`source_data`，即：

```
1 source_data=[15.0,15.8,15.2,15.1,15.9,
2              14.7,14.8,15.5,15.6,15.3,
3              15.0,15.6,15.7,15.8,14.5,
4              15.1,15.3,14.9,14.9,15.2,
5              15.9,15.0,15.3,15.6,15.1,
6              14.9,14.2,14.6,15.8,15.2,
7              15.2,15.0,14.9,14.8,15.1,
8              15.5,15.5,15.1,15.1,15.0,
9              15.3,14.7,14.5,15.5,15.0,
10             14.7,14.6,14.2,14.2,14.5];
```

设置扩充后的数据组数`num_groups`以及各组样本数据个数`samples_per_groups`：

```
1 num_groups = 25;
2 samples_per_group = 5;
```

两者相乘得到样本数据总数`num_data_points`：

```
1 num_data_points = num_groups * samples_per_group;
```

要对原始数据进行扩充，首先需要复制原始数据并添加随机扰动：

```
1 replicated_data = repmat(source_data, 1, ceil(num_data_points / ...
2       length(source_data)));
3 perturbed_data = replicated_data(1:num_data_points) + 0.05 * randn(1, ...
4       num_data_points);
```

接着使用插值生成更多数据点，并保留一位小数：

```
1 x = 1:length(perturbed_data);
2 xi = linspace(1, length(perturbed_data), num_data_points);
```

```
3 interpolated_data = interp1(x, perturbed_data, xi, 'spline');  
4 interpolated_data = round(interpolated_data, 1);
```

最后将生成的模拟数据点重新组织为25组，每组5个数据，得到完整样本数据：

```
1 expanded_data = reshape(interpolated_data, samples_per_group, ...  
    num_groups)';  
2 data = expanded_data;
```

最终的样本数据如下表所示：

	1	2	3	4	5
1	15.0000	14.6000	15.0000	15.1000	16.0000
2	14.9000	15.2000	15.5000	15.2000	14.7000
3	15.7000	14.7000	15.6000	15.2000	15.0000
4	14.2000	15.1000	15.5000	14.7000	14.6000
5	15.2000	15.5000	15.7000	14.9000	15.3000
6	14.7000	14.9000	15.1000	14.4000	14.3000
7	15.1000	15.6000	15.8000	14.8000	15.6000
8	15.9000	14.8000	15.1000	15.5000	14.3000
9	16.0000	15.3000	14.6000	15.3000	15.1000
10	15.2000	15.1000	15.1000	14.9000	14.6000
11	15.0000	14.8000	14.9000	15.1000	15.9000
12	14.9000	15.3000	15.5000	15.2000	14.7000
13	15.8000	14.8000	15.6000	15.3000	14.9000
14	14.1000	15.1000	15.4000	14.7000	14.6000
15	15.2000	15.5000	15.7000	14.9000	15.3000
16	14.6000	14.9000	15.0000	14.5000	14.3000
17	15.2000	15.6000	15.8000	14.9000	15.6000
18	15.8000	14.8000	15.1000	15.5000	14.2000
19	15.9000	15.3000	14.5000	15.2000	15.1000
20	15.2000	15.1000	15.0000	15.0000	14.5000
21	15.0000	14.7000	15.0000	15.1000	16.0000
22	14.9000	15.1000	15.6000	15.3000	14.6000
23	15.7000	14.9000	15.6000	15.4000	14.9000
24	14.2000	15.0000	15.5000	14.8000	14.7000
25	15.2000	15.4000	15.6000	14.9000	15.2000

其中每行表示一组数据，共25组，每组中有5个数据。

2.2 描述性统计分析

描述性统计量主要有以下几种：

- **均值 (Mean):** 计算每组数据的平均值，反映该组数据的中心位置。
- **方差 (Variance):** 度量数据分散程度，数值越大表示数据的波动越大。
- **标准差 (Standard Deviation):** 方差的平方根，同样反映数据的分散程度，但单位与数据相同。
- **极差 (Range):** 最大值与最小值的差，表示数据的跨度。

利用Matlab软件中自带的库函数，可以分别实现对于各组数据均值、方差、标准差与极差等统计量的计算：

```
1 means = mean(data, 2);  
2 variances = var(data, 0, 2);  
3 ranges = range(data, 2);  
4 stds = std(data, 0, 2);
```

	各组均值	各组方差	各组标准差	各组极差
1	15.14	0.268	0.51769	1.4
2	15.1	0.095	0.30822	0.8
3	15.24	0.173	0.41593	1
4	14.82	0.247	0.49699	1.3
5	15.32	0.092	0.30332	0.8
6	14.68	0.112	0.33466	0.8
7	15.38	0.172	0.41473	1
8	15.12	0.382	0.61806	1.6
9	15.26	0.253	0.50299	1.4
10	14.98	0.057	0.23875	0.6
11	15.14	0.193	0.43932	1.1
12	15.12	0.102	0.31937	0.8
13	15.28	0.187	0.43243	1
14	14.78	0.247	0.49699	1.3
15	15.32	0.092	0.30332	0.8
16	14.66	0.083	0.2881	0.7
17	15.42	0.132	0.36332	0.9
18	15.08	0.387	0.62209	1.6
19	15.2	0.25	0.5	1.4
20	14.96	0.073	0.27019	0.7

	各组均值	各组方差	各组标准差	各组极差
21	15.16	0.243	0.49295	1.3
22	15.1	0.145	0.38079	1
23	15.3	0.145	0.38079	0.8
24	14.84	0.223	0.47223	1.3
25	15.26	0.068	0.26077	0.7

以及计算总体数据的均值、方差、标准差与极差：

```
1 overall_mean = mean(data(:));
2 overall_variance = var(data(:));
3 overall_range = range(data(:));
4 overall_std = std(data(:));
```

总体均值	15.1064
总体方差	0.18657
总体标准差	0.43194
总体极差	1.9

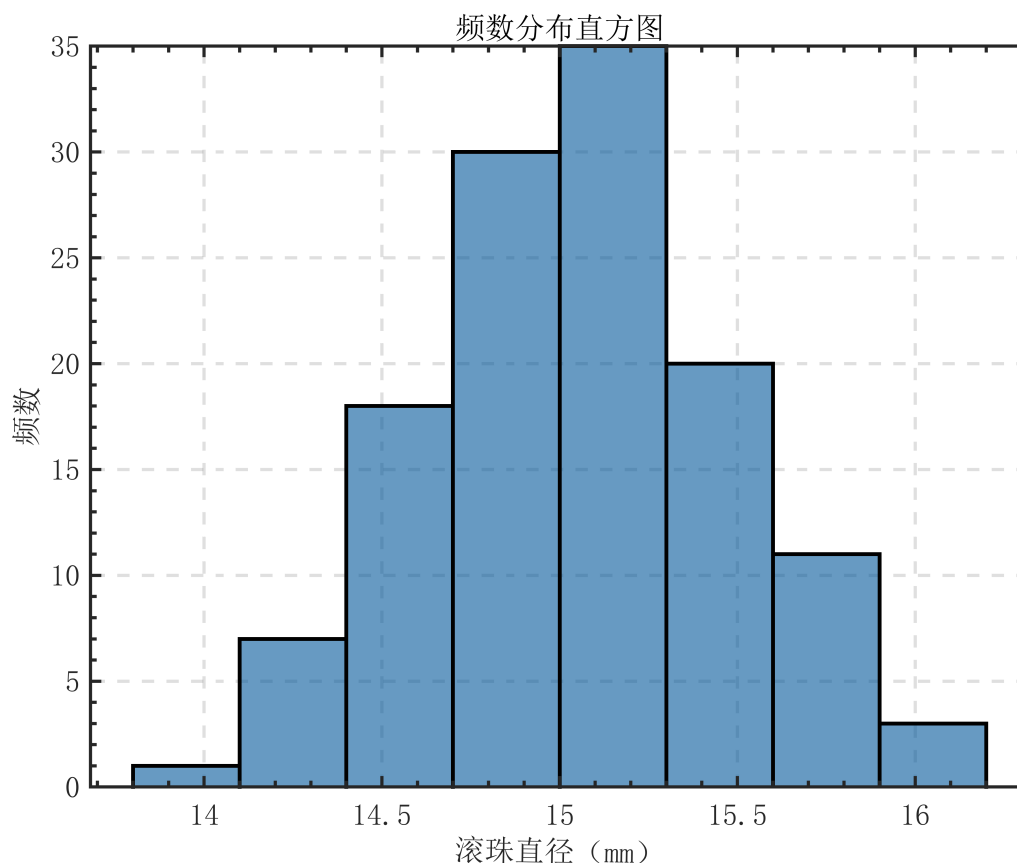
值得注意的是，Matlab中计算方差的函数 `var(A)` 默认会按照 $N - 1$ (N 是观测值数量，即上文中的 `num_data_points`) 实现归一化，从而可以作为对于总体分布的方差的无偏估计。上述得到的均为修正后的样本方差。

通过对于上述四个描述性统计量的样本观测结果进行初步分析，可以得到如下结论：

- 滚珠直径的总体平均值约为15.1mm，可作为后续总体均值检验的检验目标。
- 各组数据的方差均在0.5以内，总体方差与标准差也较低，说明生产过程中数据波动较小，产品质量稳定性较高。
- 数据极差范围适中，总体极差不超过2，处于合理区间。

除此之外，还可以通过绘制频数分布直方图的方式对于样本数据进行描述性统计分析：

```
1 h = histogram(data);
```



观察图表可知，数据的频数分布大致呈“两边少，中间多”的趋势，推测其总体大致服从正态分布，需要进行进一步检验。

2.3 数据正态性检验

要验证总体是否服从正态分布，需要采用Pearson卡方检验方法，对总体的分布类型进行检验。

Matlab软件中，有相应的函数`chi2gof`可通过Pearson卡方检验来验证总体分布类型是否为正态分布：

```
1 [result, p_chi2] = chi2gof(data(:));
```

`chi2gof`函数是使用Pearson卡方检验返回原假设的检验决策，其原假设假定样本数据来自正态分布的总体，备择假设是样本数据不来自正态分布的总体。默认情况（仅将样本数据`data`输入`chi2gof`函数）下，检验的显著性水平为 $\delta = 0.05$ ，且默认检验总体是否服从正态分布；通过增加函数的输入参数可以对显著性水平和检验的分布类型进行更改，在此不再赘述。

返回的参数`result`为正态性检验的结果，若`result = 0`，说明无法拒绝原假设，即可认为样本数据来自正态分布的总体；若`result = 1`，则拒绝原假设，认为数据不来自正

态分布的总体。另一返回参数 p_chi2 则为该检验的 p 值，当其大于显著性水平 δ 时不拒绝原假设。

通过对样本数据进行正态性检验，可认为数据来自正态分布的总体，正态性检验的 p 值为0.37005。

从原理出发，也可根据假设检验的基本方法流程对总体的分布类型进行检验：

首先，提出原假设与备择假设：

- H_0 ：总体服从正态分布，即 $F(x) = \phi(\frac{x-\mu}{\sigma})$ ；
- H_1 ：总体不服从正态分布，即 $F(x) \neq \phi(\frac{x-\mu}{\sigma})$ 。

其中 $F(x)$ 为总体的分布， μ 和 σ 为正态分布的均值与方差（均为未知参数）。

接着，根据假设的提出给出对应的拒绝域形式： $R_0 = \{T > c\}$ ，其中 T 为构造的统计量：

$$T = \sum_{i=1}^r \frac{(n_i - np_i)^2}{np_i} \quad (1)$$

若将对总体 X 所有可能的取值范围分割成 r 个两两不交的部分，则式中：

- n_i ：统计样本中实际落入第 i 个部分的个数（实际频数）；
- n ：样本数据总数，即上文代码中的`num_data_points`；
- p_i ：任一样本落入第 i 个部分的概率。

显然，当原假设 $H_0 : F(x) = \phi(\frac{x-\mu}{\sigma})$ 成立时，理论频数 $n * p_i$ 应与实际频数 n_i 较为接近，这也是拒绝域形式的确定依据。

可以注意到，Pearson卡方统计量 $T = \sum_{i=1}^r \frac{(n_i - np_i)^2}{np_i}$ 服从如下的大样本分布：

$$\hat{T} = \sum_{i=1}^r \frac{(n_i - n\hat{p}_i)^2}{n\hat{p}_i} \rightarrow X^2(r - 1 - s) \quad (2)$$

其中：

- s 表示待估未知参数的个数，显然本项目中 $s = 2$ ；
- \hat{p}_i 为任一样本落入第 i 部分的概率 p_i 的估计值，这是由于假定的分布类型中含有未知参数，因此需要先根据样本数据对未知参数进行极大似然估计，再代入得到概率的估计值。

然后，需要计算犯第一类错误的概率 $\alpha = P\{H_1|H_0\} = \{T > c|F(x) = F_0(x)\}$ ，并根据 $\alpha = \delta = 0.05$ 求得拒绝域中未知参数 c 的临界值 $criticalValue = 11.0705$ ：

```

2 mu_MLE = overall_mean;
3 sigma2_MLE = overall_variance*(num_data_points-1)/num_data_points;
4 sigma_MLE = sqrt(sigma2_MLE);
5 dimension=numBins-1-2;
6 criticalValue = chi2inv(1 - deta, dimension);

```

事实上，上文中提到Matlab软件中的`var(A)`函数默认按照 $N - 1$ 实现归一化，而正态分布方差的极大似然估计应为未修正的样本方差，故特此进行更正；同时，在2.2节的描述性统计分析过程中，涉及到频数直方图的绘制时已经对数据完成了分组，可直接通过读取上述的图表属性获取分组数`numBins`（对应上文中 r ）、各组频数`binCounts`（对应上文中实际频数 n_i 与各分组边界`binEdges`（可用于计算 p_i ））：

```

1 numBins = h.NumBins;
2 binCounts = h.Values;
3 binEdges = h.BinEdges;

```

最后，通过样本数据计算出对应卡方统计量 T 的观测值 $kafang = 2.6897$ ：

```

1 kafang=0;
2 p_hat(1)= normcdf(binEdges(2), mu_MLE, sigma_MLE);
3 for i=2:numBins-1
4     p_hat(i)= normcdf(binEdges(i+1), mu_MLE, ...
5         sigma_MLE)-normcdf(binEdges(i), mu_MLE, sigma_MLE);
6 end
7 p_hat(numBins)= 1-normcdf(binEdges(numBins), mu_MLE, sigma_MLE);
8 for i=1:numBins
9     kafang=kafang+(binCounts(i)-num_data_points*p_hat(i))^2 / ...
10         (num_data_points*p_hat(i));
11 end

```

由于 $kafang < criticalValue$ ，即不位于拒绝域内，因此无法拒绝原假设 H_0 ，即可以认为数据确实来自正态分布的总体，与直接调用Matlab内置库函数的结果一致。

2.4 总体均值检验

通过Pearson卡方检验，已经可以认为在显著性水平 $\delta = 0.05$ 的情况下，总体近似服从正态分布。进一步地，还希望对该正态总体的均值参数 μ 进行假设检验。在2.2节的描述性统计分析过程中，已经计算得到样本的总体均值（实际上也是正态分布参数 μ 的无偏估计与最大似然估计） μ_0 约为15.1(mm)，因此接下来的假设检验过程就将其作为检验对象：

```
1 miu0=15.1;
```

在正态总体的均值检验过程中，主要采取t检验的方法，即认为正态总体的另一参数 σ 未知。这样的检验方法的合理性是显然的，此时需要构造服从t分布的统计量以进行t检验。

在Matlab软件中，对于t检验仍然有相对应的库函数`ttest()`可以调用：

```
1 [result1, ttest_p_value] = ttest(data(:),miu0);
```

函数`ttest(data(:),miu0)`使用单样本t检验返回原假设的检验决策，其原假设假定数据`data`来自均值等于输入参数`miu0`且方差未知的正态分布，而备择假设是总体分布的均值不等于`miu0`。如果该检验在默认为5%的显著性水平上拒绝原假设，则函数的返回值`result1`为1，即认为正态总体的均值不为`miu0`，否则为0；而另一返回值`ttest_p_value`则为该检验的 p 值，当其大于显著性水平时不拒绝原假设，即可认为正态总体的均值等于`miu0`。

运行代码可得：正态总体均值检验 p 值`ttest_p_value = 0.8687`，可认为正态总体的均值约为 $\mu_0 = 15.1$ 。

同样的，这样的假设检验过程也可以基于一般的统计方法实现：

首先，提出原假设与备择假设：

- H_0 ：正态总体的均值为15.1，即 $\mu = \mu_0 = 15.1$ ；
- H_1 ：正态总体的均值不为15.1，即 $\mu \neq \mu_0$ 。

接着，根据假设的提出给出对应的拒绝域形式： $R_0 = \{(x_1, x_2, \dots, x_n)^T : |\bar{x} - \mu_0| > c\}$ 。该拒绝域形式的构造是具有合理性的，因为显然可证明样本均值 \bar{x} 是对于正态总体参数 μ 的无偏估计，若原假设 H_0 成立，则说明 \bar{x} 应与设定检验值 μ_0 较为接近，反之可得其拒绝域应描述为 \bar{x} 与 μ_0 相距较远的情形，基于这样的原理也可构造其他形式的拒绝域，这里只是给出一种可能性。

然后，需要计算犯第一类错误的概率 $\alpha = P\{H_1|H_0\} = \{|\bar{x} - \mu_0| > c | \mu = \mu_0\}$ ，并根据 $\alpha = \delta = 0.05$ 求得拒绝域中未知参数 c 的临界值`tCriticalValue = 0.0765`：

```
1 deta=0.05;
2 dimension=num_data_points-1;
3 s=sqrt(var(data(:)));
4 tCriticalValue = tinv(1 - deta/2, dimension)*s/sqrt(num_data_points);
```

上述计算操作的依据是，在对第一类错误发生概率 α 的计算过程中，可以根据抽样分布定理构造出对应的t统计量 $\frac{\bar{x}-\mu_0}{s/\sqrt{n}} \sim t(n-1)$ 从而查表进行计算求解得到临界值 $c =$

$\frac{t_{1-\frac{\delta}{2}}(n-1)*s}{\sqrt{n}}$ ，其中 s 为样本数据的标准差（修正后）， n 为样本容量 num_data_points 。

最后，前文已经计算出样本均值为 \bar{x} ，若将其与待检验值 μ_0 之间的距离 $|\bar{x} - \mu_0|$ 定义为参数 d ，则在判定是否处于拒绝域中时只需要比较 d 与临界值 $tcriticalValue$ 即可：

```
1 d=abs(overall_mean-miu0);
```

计算得到 $d = 0.0064$ ，小于临界值 $tcriticalValue = 0.0765$ ，即不位于拒绝域内，因此无法拒绝原假设，可认为正态总体的均值 μ 约为**15.1**。这同样与直接调用Matlab内置函数的结果相一致。事实上，只要 μ_0 在样本均值 $\bar{x} = 15.1064$ 上下 $tcriticalValue = 0.0765$ 的区间范围内，都可在显著性水平 $\delta = 0.05$ 的情况下认为是正态总体的均值 μ 。

2.5 数据工序能力指数的计算与评估

在2.3节和2.4节中，通过假设检验的方法已经分别证明在显著性水平 $\delta = 0.05$ 的情况下，可以认为总体近似服从正态分布，且正态总体的均值 μ 约为**15.1mm**。基于类似的方法，不难得到正态总体的标准差 σ 约等于样本数据的标准差（修正） $s = 0.4319$ 。基于这样的假设并通过假设检验验证其合理性之后，可以对于产品的生产质量利用数据工序能力指数等指标进行评估。

在工业生产中，为了综合表示工艺水平满足工艺参数规范要求的程度，通常借助于工序能力指数 (Process Capability index, Cp or Cpk) 来描述生产线是否具有较高的工艺水平能否生产出质量好的产品。潜在工序能力指数 C_p 的定义为：

$$C_p = \frac{T_U - T_L}{6\sigma} \quad (3)$$

其中 T_U ， T_L 分别表示工艺参数规范的上限和下限。在本项目的问题背景之下，一般认为滚珠直径在**14mm-16mm**都可视为合理；但值得注意的是，在此定义中，隐含要求工艺参数分布中心 μ 与工艺规范要求的中心值 $T_0 = (T_U + T_L)/2$ 相重合。由2.4节的均值检验过程可知， μ 约为**15.1**且上下限**14**和**16**的均值**15**不在可以通过假设检验的区间范围内，因此在此先进行近似处理，假定 $T_U = 16.1$ ， $T_L = 14.1$ ，则可以计算出**此时的潜在工序能力指数** $C_p = 0.7717$ ：

```
1 Tu=16.1;
2 Tl=14.1;
3 spec_limits = [Tl, Tu];
4 sigma = std(data(:));
5 cp = (spec_limits(2) - spec_limits(1)) / (6 * sigma);
```

而事实上，在原设定 $T_U = 16$ ， $T_L = 14$ 的条件下，由于规范中心 $T_0 = (T_U + T_L)/2$ 与

参数分布中心 μ 不重合，上述的近似处理并不能很准确的反映其工序能力，此时则需要采用实际工序能力指数 C_{pk} 来度量工艺水平的高低：

$$C_{pk} = \frac{T_U - T_L}{6\sigma}(1 - K) \quad (4)$$

其中 K 用于度量工艺参数分布均值 μ 对规范中心 T_0 的相对偏离度，即：

$$K = \frac{|\mu - T_0|}{(T_U - T_L)/2} \quad (5)$$

```
1 Tu=16;
2 Tl=14;
3 spec_limits = [Tl, Tu];
4 T0=(spec_limits(2) + spec_limits(1)) / 2;
5 k=abs(overall_mean-T0)*2/(spec_limits(2) - spec_limits(1));
6 cpk=(spec_limits(2) - spec_limits(1)) *(1-k)/ (6 * sigma);
```

计算得到：相对偏离度 $K = 0.1064$ ，实际工序能力指数 $C_{pk} = 0.6896$ 。其中相对偏离度 $K < 1$ ，说明均值未偏离到规范范围外，表明工艺加工结果合适，且实际工序能力指数 $C_{pk} > 0$ ，表明该工序具有一定的生产能力，但并未达到国际上的普遍要求 $C_{pk} > 1.5$ ，说明生产水平还存在相当大的提升空间。

除此之外，还可以计算单侧规范下的实际工序能力指数 C_{PU} 和 C_{PL} ：

若工艺参数只有上规范 T_U 的要求，则实际工序能力指数 C_{PU} 定义为：

$$C_{PU} = \frac{T_U - \mu}{3\sigma} \quad (6)$$

若工艺参数只有下规范 T_L 的要求，则实际工序能力指数 C_{PL} 定义为：

$$C_{PL} = \frac{\mu - T_L}{3\sigma} \quad (7)$$

```
1 cpu = (spec_limits(2) - mean(data(:))) / (3 * sigma);
2 cpl = (mean(data(:)) - spec_limits(1)) / (3 * sigma);
```

计算得到：实际工序能力指数 $C_{PU} = 0.6896$ 、 $C_{PL} = 0.8538$ ，也并未达到国际一般标准。

2.6 均值控制图与方差控制图

Shewhart控制图是实施SPC过程中判断生产过程是否处于统计受控状态的基本工具，其中使用最为广泛的为均值控制图和方差（标准差）控制图。通过连续采集工艺参数数据，可以利用Shewhart控制图来定量分析和度量生产线的运行过程处于统计受控状态。

2.6.1 均值控制图

在均值控制图中，包含有三条水平线，分别是：

- **UCL(Upper Control Limit)**: 表示上控制限；
- **LCL(Lower Control Limit)**: 表示下控制限；
- **CL(Center Line)/Avg(Average)**: 表示中心线。

对于各批次的样本数据，用折线连接每批次的均值，从而得到反映不同批次数据均值波动情况的折线图。

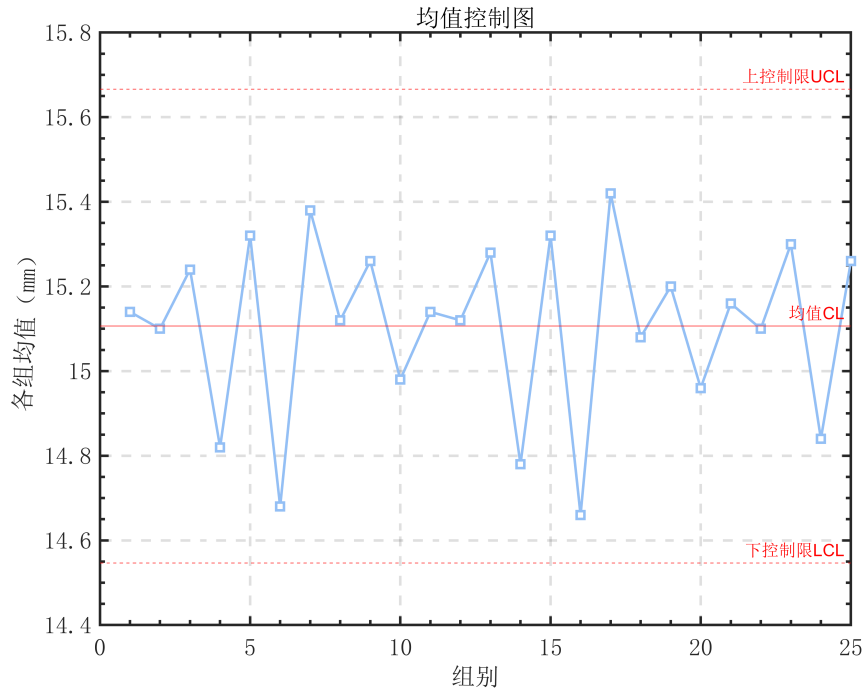
对于服从正态分布总体 $N(\mu, \sigma^2)$ 的工艺参数而言，其取值落在 $(\mu \pm 3\sigma)$ 区间范围内的概率为99.73%，因此可以采用如下的 3σ 准则来确定均值控制图的中心线以及上下控制限：

- $UCL = \mu + 3\sigma$
- $CL = \mu$
- $LCL = \mu - 3\sigma$

* 我国标准“GB/T 4091常规控制图(2001版)”规定采用上述法则来计算控制限。

基于以上法则，可绘制均值控制图如下：

```
1 miu_hat=overall_mean;
2 sigma_hat=overall_variance;
3 figure;
4 plot(means, 'o-');
5 CL= miu_hat;
6 UCL=miu_hat+3*sigma_hat;
7 LCL=miu_hat-3*sigma_hat
8 yline(CL, 'r-', 'CL');
9 yline(UCL, 'r--', 'UCL');
10 yline(LCL, 'r--', 'LCL');
```



但在实际生产实践中，并不是所有的产品工艺指标都会近似服从正态分布。因此，在实际操作中，均值控制限常用如下公式估算：

- $UCL = \hat{\mu} + A_s \cdot \hat{s}$
- $CL = \hat{\mu}$
- $LCL = \hat{\mu} - A_s \cdot \hat{s}$

其中，若采集工艺参数样本数据组数 $k = 25$ ，各批次数据数量 $n = 5$ ，令 x_{ij} 表示第 i 批次的第 j 个数据， \bar{x}_i 表示组内均值；则可用 μ 表示个各批次所有数据的样本均值， \hat{s} 表示各个批次数据标准偏差的算数平均值，即：

$$\bar{x}_i = \frac{1}{n} \sum_{j=1}^n x_{ij} \quad (8)$$

$$\hat{s} = \frac{1}{k} \sum_{i=1}^k \hat{s}_i \quad (9)$$

$$\hat{\mu} = \frac{1}{k} \sum_{i=1}^k \bar{x}_i = \frac{1}{kn} \sum_{i=1}^k \sum_{j=1}^n x_{ij} \quad (10)$$

$$\hat{s}_i = \sqrt{\frac{1}{n-1} \sum_{j=1}^n (x_{ij} - \bar{x}_i)^2} \quad (11)$$

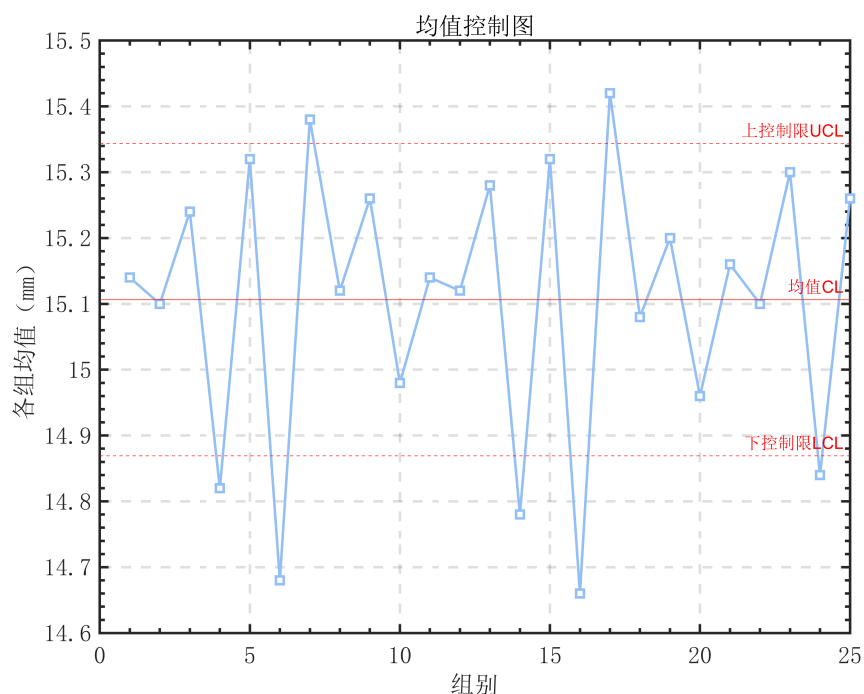
又查表^[2]可得：当每批次样本数据数 $n = 5$ 时，有 $A_s = 1.427$ 。

于是绘制出实际均值控制图:

```

1 figure;
2 plot(means, 'o-');
3 xi=mean(data, 2);
4 for i=1:num_groups
5     sum=0;
6     for j=1:samples_per_group
7         sum=sum+((data(i,j)-xi(i))^2);
8     end
9     si_hat(i)=sqrt(sum/(num_groups-1));
10 end
11 s_hat=mean(si_hat);
12 As=1.427;
13 CL= miu_hat;
14 UCL=miu_hat+s_hat*As;
15 LCL=miu_hat-s_hat*As;
16 yline(CL, 'r-', 'CL');
17 yline(UCL, 'r--', 'UCL');
18 yline(LCL, 'r--', 'LCL');

```



2.6.2 方差控制图

方差控制图的结构与均值控制图非常类似,也是由中心线以及上下控制限构成。只是图中的数据点描述的每批次数据的标准偏差,因此可用于定量判断不同批次数据的标准偏差(数据的波动性)在变化中是否存在“异常因素”。

类似于均值控制图，在实际操作中，常用如下公式估算其中心线以及上下控制限：

- $UCL = B_U \bullet \hat{s}$
- $CL = \hat{s}$
- $LCL = B_L \bullet \hat{s}$

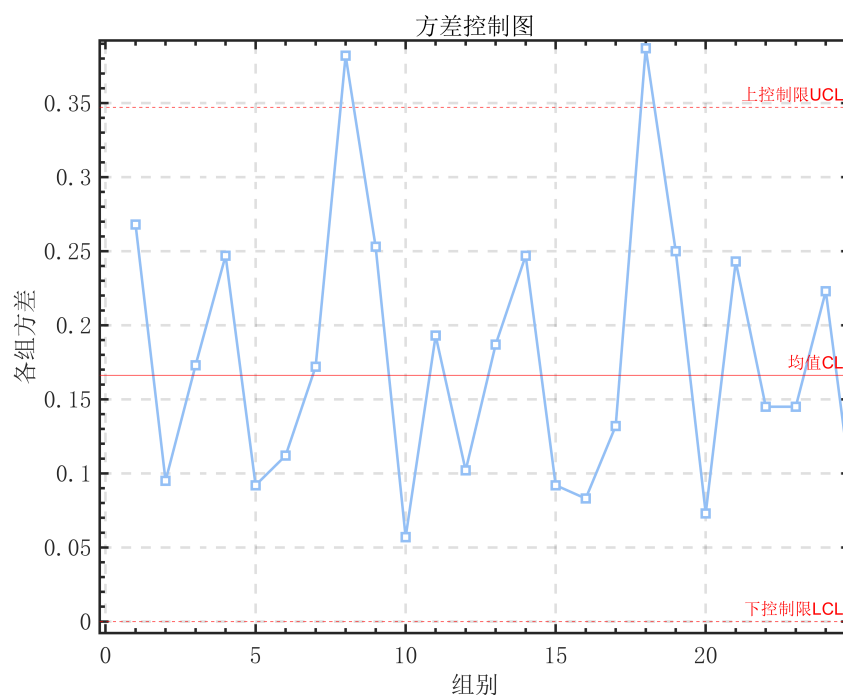
其中 B_U 、 B_L 为依赖于每批次样本数据数 n 的常数参数，查表^[2]可得：当每批次样本数据数 $n = 5$ 时，有 $B_U = 2.089$ 、 $B_L = 0$ 。

据此可绘制出对应的方差控制图：

```

1 Bu=2.089;
2 Bl=0;
3 figure;
4 plot(variances, 'o-');
5 hold on;
6 UCL=(Bu*s_hat);
7 CL=(s_hat);
8 LCL=(Bl*s_hat);
9 yline(CL, 'r-', 'CL');
10 yline(UCL, 'r--', 'UCL');
11 yline(LCL, 'r--', 'LCL');

```

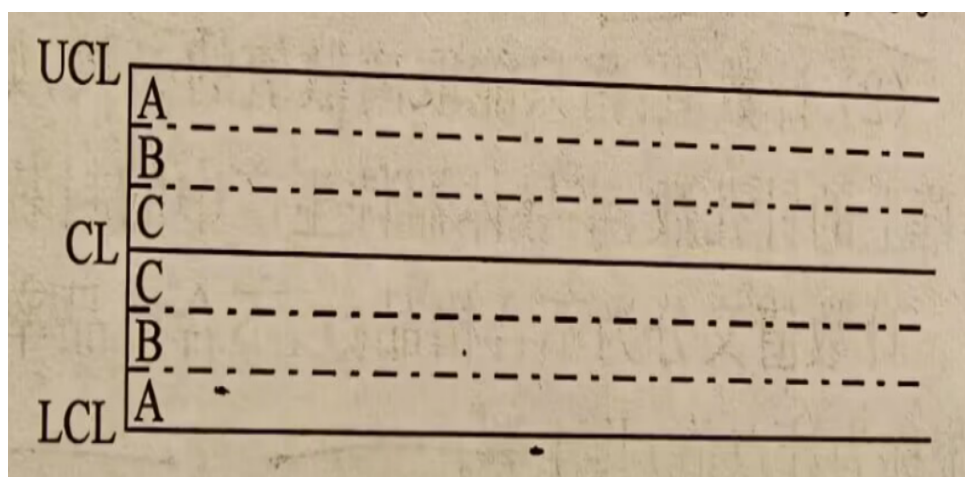


3 产品质量分析结论

为根据Shewhart控制图判断产品生产过程是否受控，可对于均值控制图与方差控制图分别采用如下八条常见规则（规则均不能触碰，否则视为不受控），以识别产品生产过程是否仅受到随机因素影响：

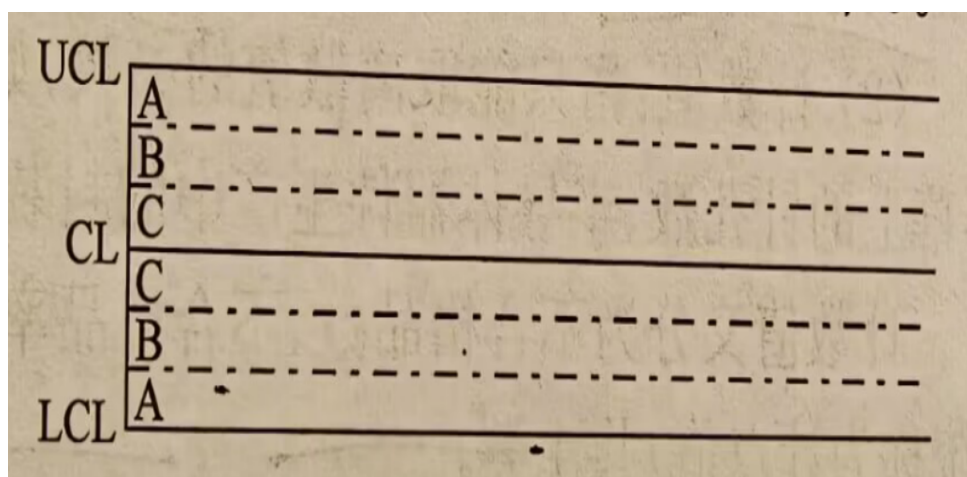
- 规则 1:控制图上有一个点(对应某个批次数据)位于控制限以外；
- 规则 2:连续9个点落在中心线同一侧；
- 规则 3:连续6个点递增或者递减；
- 规则 4:连续14个点交替上下；
- 规则 5:连续3个点中有2个点落在中心线同侧的B区以外；
- 规则 6:连续5个点中有4个点落在中心线同侧的C区以外；
- 规则 7:连续15个点落在中心线两侧的C区；
- 规则 8:连续8个点落在中心线两侧且无一点在C区以内。

其中，A、B、C分区如下图^[2]所示，C区为以CL控制限为对称轴的中心区域，B、A两区向两侧延伸至上下控制限UCL与LCL。



实际上，Matlab软件中对于Shewhart控制图以及SPC统计过程控制方法有一套完整封装的库函数`controlchart()`，调用结果如下所示：

```
1 load parts
2 [st,plotdata] = controlchart(data,'charttype',{'xbar' 's'});
```



其中第一张图为均值控制图，第二张图为标准差控制图。

通过观察图像，基于八大规则进行判断并通过查看返回对象`plotdata`的`ooc`属性（Logical that is true for points that are out of control）可知，该产品生产过程确实处于统计受控状态。除此之外，样本均值也在15.1mm左右，符合对于滚珠直径的一般行业要求，工艺水平良好；但实际工序能力指数 C_{pk} 仅有0.6896，说明工艺水平还有上升空间。

参考文献

- [1] 茆诗松,程依明,濮晓龙编著. 概率论与数理统计教程[M]. 高等教育出版社, 2019.
- [2] 贾新章 .. [等] 编著. 统计过程控制理论与实践[M]. 电子工业出版社, 2017.